

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ  
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
ФОРМА НАВЧАННЯ ДЕННА  
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ  
ІНФОРМАТИКИ**

**Допускається до захисту**

Завідувач кафедри \_\_\_\_\_ О.О. Ємець  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2019 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО ДИПЛОМНОЇ РОБОТИ**

**на тему**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ТРЕНАЖЕРА З ТЕМИ «АЛГЕБРА  
ПРЕДИКАТИВ» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ  
«МАТЕМАТИЧНА ЛОГІКА»**

**зі спеціальності 122 «Комп'ютерні науки»**

**Виконавець роботи** Куркін Володимир Владиславович

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2019р.  
(підпис)

**Науковий керівник** к.ф.-м.н., доц. Черненко Оксана Олексіївна

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2019р.  
(підпис)

**ПОЛТАВА 2019р.**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	3
ВСТУП .....	4
1. ПОСТАНОВКА ЗАДАЧІ.....	6
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
2.1. Переваги і недоліки дистанційного навчання.....	8
2.2. Огляд тренажерів зі схожою тематикою .....	13
2.3. Переваги та вади розглянутих тренажерів .....	20
2.4. Актуальність теми роботи.....	21
3. ТЕОРЕТИЧНА ЧАСТИНА .....	24
3.1. Алгебра предикатів .....	24
3.2. Приклади застосування алгебри предикатів .....	29
3.3. Алгоритмізація за темою роботи.....	31
3.4. Розробка блок-схеми.....	41
4. ПРАКТИЧНА ЧАСТИНА .....	52
4.1. Обґрунтування вибору програмних засобів.....	52
4.2. Опис процесу програмної реалізації .....	56
4.3. Опис програми.....	62
4.4. Інструкція по використанню тренажеру.....	66
ВИСНОВКИ.....	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	72
ДОДАТОК А. КОД ПРОГРАМИ .....	<b>Ошибка! Закладка не определена.</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
логіка предикатів	розділ класичної символічної логіки, що вивчає суб'єктно-предикатну структуру висловлювань, на підставі чого визначають значення істинності висловлювань
предикат	функція з областю значень $\{0,1\}$ (або «Істина» та «Хибна»), певна на $n$ -й декартовій ступені множини $M$ .
$n$ -місний предикат $P(x_1, \dots, x_n)$	деяка логічна функція $n$ змінних $x_1, \dots, x_n$ , що визначена на множині $\Omega$ і приймає значення істина або хибне
квантор загальності	висловлювання, в якому властивість $P$ приписують певному непорожньому класу загалом, що означає: для всіх елементів класу $A$ притаманна властивість $P$
квантор існування	висловлювання про певний непорожній клас, в якому властивість $P$ притаманна лише декотрим елементам цього класу, тобто існують елементи класу $A$ , яким притаманна властивість $P$
JavaScript (JS)	динамічна, об'єктно-орієнтована прототипна мова програмування

## ВСТУП

Сучасний рівень розвитку комп'ютерної техніки і програмного забезпечення надає широкі можливості щодо модернізації та підвищення ефективності навчання. Використання мультимедійних технологій у навчальному процесі урізноманітнює і автоматизує його, підвищує ефективність засвоєння матеріалу, здійснює контроль знань. За останні роки розвиток і зростання мережі Інтернет, її вплив на діяльність суспільства, зумовили істотні структурні зміни в освітній системі. У зв'язку з цим можна стверджувати, що впровадження інформаційних технологій у повсякденне життя українських громадян стало поштовхом для розвитку процесів пов'язаних з дистанційним навчанням.

Мета роботи – алгоритмізація та програмна реалізація тренажера з теми «Алгебра предикатів» дистанційного навчального курсу «Математична логіка» та закріплення знань з використання алгебри предикатів.

Об'єктом розробки в даній роботі є процес дистанційного навчання математичним дисциплінам.

Предметом розробки є програмний продукт, що реалізує тренажер з теми «Алгебра предикатів» на мові програмування JS.

Головне завдання – розробити алгоритм роботи тренажера з теми «Методи перевірки тотожної істинності логічних формул» та створити її програмну реалізацію.

До методів розробки відноситься методика застосування алгебри предикатів першого порядку, мова програмування JavaScript (JS) і фреймворк AngularJS.

Для викладання даного тренажера студентам іноземцям реалізована можливість зміни мови між українською та англійською.

Тренажер готовий до використання в дистанційному курсі «Математична логіка».

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі для реалізації тренажера. У другому розділі описано переваги і недоліки дистанційного навчання, огляд тренажерів зі схожою тематикою та їх особливості, актуальність теми роботи. У третьому розділі представлено теоретичний матеріал з алгебри предикатів, наведено приклади застосування алгебри предикатів, описано алгоритм роботи тренажеру та представлено блок-схему алгоритму. У четвертому розділі – описано обґрунтування вибору програмних засобів, процес програмної реалізації, основні елементи програми і інструкцію по використанню тренажера.

Обсяг пояснювальної записки: 78 стор., в т.ч. основна частина - 63 стор., джерела - 15 назв.

## 1. ПОСТАНОВКА ЗАДАЧІ

Розглянемо основні завдання роботи:

- вказати актуальність теми;
- розглянути теоретичні відомості про алгебру предикатів, логіку першого порядку;
- розглянути специфіку застосування алгебри предикатів;
- розробити алгоритм тренажера з теми «Алгебра предикатів»;
- створити програмну реалізацію алгоритму;
- описати процес створення програми;
- зробити опис розробленої програми;
- написати інструкцію користувача.

Оскільки тренажер створюється для дистанційного курсу, то потрібно описати переваги і недоліки дистанційного навчання, розглянути тренажери схожої тематики, вказати їх переваги та вади.

В алгоритмі слід розробити можливість студента звернутися до теоретичного матеріалу з теми, що допоможе йому у проходженні завдання та покроково описати всі дії при розв'язанні наведеного прикладу [2].

Розроблений тренажер повинен мати зрозумілий для користувача інтерфейс.

Для цього він повинен містити наступні елементи:

- головну сторінку;
- можливість ознайомитися з теоретичним матеріалом;
- виведення умови прикладу та завдання;
- варіанти відповідей до завдання;
- виведення повідомлення про помилку;
- можливість завершити роботу програмного продукту.

Для викладання даного тренажера студентам іноземцям необхідно реалізувати можливість зміни мови між українською та англійською. Теоретичний матеріал теж має бути переведений на дві мови.

При програмній реалізації потрібно розробити виведення повідомлення про помилку у разі неправильної відповіді. Це повідомлення має містити правильну відповідь.

Результати роботи мають бути опубліковані в збірнику наукових статей та матеріалах науково-практичного семінару.

## 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

### 2.1. Переваги і недоліки дистанційного навчання

До традиційних видів освіти в Україні належить очна, заочна, суботня форми навчання та екстернат. До форм очної освіти належить денна ( очна) та вечірня форми освіти. Їх особливостями є:

- навчальний процес, заснований на двосторонній взаємодії між тим, хто навчається і тим, хто навчає;
- використовуються групові форми навчальних занять: лекція, семінар, лабораторна робота, практична робота, колоквиум, консультації, тощо;
- чітко заданий перелік дисциплін, що вивчаються;
- чітке кредитування навчального часу.

Своєю чергою, заочна форма освіти розрахована на осіб, які вже здобули одну вищу освіту, працюють або з певних причин не можуть відвідувати заняття щоденно (як у випадку із очною формою навчання). Така форма навчання забезпечує:

- можливість отримання освіти незалежно від місця проживання;
- одночасного навчання за декількома спеціальностями;
- можливість отримання вищої освіти різними категоріями населення ( непрацездатні, інваліди, жінки, які виховують дітей тощо);
- одночасне охоплення великої кількості студентів;
- навчання за власним графіком у міжсесійний період;
- можливість практично негайно застосовувати свої знання на практиці, не закінчивши навчання.

Зрозуміло, що в основі дистанційної освіти закладені принципи та особливості традиційних форм навчання, окрім того, додані нові особливості, які полягають у використанні інтернет-технологій для



доступу для навчальних матеріалів, інтерактивної взаємодії між студентами тощо.

Так, принцип спрямування навчання на вирішення завдань освіти і загального розвитку студентів в традиційній освіті означає, що викладач повинен звертати увагу не лише на вирішення завдань та вмінь, а й на ефективність проведеної системи виховних заходів у розглянутій темі. У дистанційній системі навчання цей принцип отримав таку інтерпретацію: принцип креативності характеру пізнавальної діяльності. За допомогою інтерактивних технологій креативний характер дистанційного навчання може реалізуватись за рахунок суперництва та змагання великої кількості студентів, що підвищує їхній творчий потенціал.

Одним із найважливіших принципів у традиційній формі освіти є принцип науковості. Він ґрунтується на зв'язку між наукою та предметом, що вивчається, вимагає, щоб зміст матеріалів навчання забезпечував інформацію про наукові факти, поняття, закономірності, сучасні досягнення та відкриття. Більш фундаментальну форму цей принцип отримав у дистанційному навчанні: дидактичний принцип відповідності фундаментальності навчання пізнавальним потребам особи, яка навчається. Цей принцип полягає у тому, що навчання вважається фундаментальним, якщо воно орієнтується на визначення основ та залежностей між різноманітними процесами навколишнього середовища. Разом з цим, принцип висуває такі вимоги до студента, як:

- висока мотиваційна потреба;
- прагнення до власного розвитку;
- корекції та самокритики;
- спрямованість особистості до досягнення поставленої мети.

Продовженням змісту цього принципу є принцип вільного вибору інформації, яка отримується, шляхом визначеної діяльності: не існує єдиного ідеального інформаційного джерела, тому спрямованість навчання стосується безпосередньо не інформації, а шляхів її перетворення та

опрацювання, за допомогою участі в дискусіях, телеконференціях, роботи з пошуковими машинами тощо.

Наступним, не менш важливим дидактичним принципом у традиційних формах навчання є принцип систематичності та послідовності, який потребує того, щоби знання, які подаються, були впорядковані, класифіковані, логічно пов'язані з іншим матеріалом, що призводить до покращення результатів навчання. Натомість, у дистанційному навчанні надається можливість самостійно обирати навчальні цілі, форму та темп навчання. Такий принцип отримав назву принципу індивідуальної навчальної траєкторії студента.

У принципі наочності зазначено, що знання засвоюються краще, коли наявна висока концентрація сприйняття всіма органами чуття людини. Оскільки у дистанційному навчанні відсутній безпосередній контакт аудиторії та викладача, сформульовано принцип віртуалізації навчання. У дистанційній формі навчання широко використовуються мультимедійні видання, відеоролики, ілюстраційні матеріали, відеоконференції тощо.

Окрім описаних вище принципів, також існує група принципів дистанційного навчання, створення яких є наслідком активного розвитку та використання інформаційних технологій. Нижче розглянемо деякі із цих принципів.

Так, принцип ідентифікації є важливим з огляду на те, що у дистанційному навчанні існує більше можливостей фальсифікації навчання, ніж у традиційній очній формі навчання. Контролювати самостійність виконання контрольних заходів можна за допомогою використання технічних засобів, наприклад, відеозв'язку.

Інформаційні технології повинні забезпечувати можливість контролю викладачем навчального процесу, можливість вносити зміни в навчальний курс, робити доступним контакт не лише викладача та

студента, а і забезпечувати можливість контактів студентів між собою – це принцип інтерактивності.

Для ефективного дистанційного навчання важливим є принцип початкових знань, який полягає у тому, що користувач дистанційного курсу повинен володіти навичками роботи на комп'ютері, мати доступ до Інтернету, обов'язковими також є навички роботи в мережі та необхідне технічне забезпечення для повноцінного навчання.

Вагомим педагогічним принципом є принцип педагогічної доцільності застосування засобів інформаційних технологій. Кожен крок проектування та організації процесу дистанційного навчання вимагає педагогічного оцінювання та оцінювання доцільності використання новітніх інформаційних технологій, які мають безпосередній вплив на компоненти навчання – його зміст, мету, засоби тощо.

Зазначимо, що структура наведених вище педагогічних принципів не є сталою – з часом цілком реальні зміни та нововведення, пов'язані із подальшим розвитком та впровадженням дистанційного навчання.

Однак, власне ці принципи є визначальними при побудові систем дистанційного навчання. Складовими цих систем є велика кількість модулів, зв'язаних між собою. При цьому є виділені основні функціональні можливості, які повинні забезпечувати сучасні системи дистанційного навчання. Цей список здебільшого схожий у різних системах, які існують сьогодні.

- **Доступ до навчального контенту.** Необхідно забезпечити можливість авторизації користувача, управління правами груп користувачів, контроль доступу до навчальних матеріалів.

- **Забезпечення зручних засобів адміністрування.** Типовим набором функціональних можливостей є реєстрація користувачів, управління групами користувачів, управління дистанційними курсами та заходами контролю тощо.

- **Надання засобів комунікації між користувачами курсу.** Сьогодні існує багато можливостей забезпечення зв'язку, до основних належать відео- та аудіоконференції, форуми, чати, блоги, електронна пошта тощо.

- **Можливість формування навчальних планів із використанням моделей управління компетенціями.** Для розподілення навчальних планів будуються моделі, які вказують роль, яку виконує користувач та набір його можливостей ( компетенцій) відповідно до заданої ролі. В подальшому для кожної компетенції є набір курсів, які потрібно вивчити і набір контрольних заходів, які потрібно скласти. На основі побудованих моделей система створює відповідні навчальні плани для кожного користувача.

- **Формування звітів.** Важливим є забезпечення можливості формування звітів за побажанням користувача.

- **Інтеграція системи з різноманітними інформаційними системами.** Система не повинна бути ізольованою від інших інформаційних систем, з якими знаходиться у загальному середовищі.

- **Формування складних розподілених систем.** Необхідно забезпечити спрощення доступу користувачів до навчальних курсів за рахунок побудови системи дистанційного навчання із декількома вузлами. Важлива присутність автономного клієнта, який забезпечить доступ до матеріалів за відсутності постійного доступу до системи навчання [3].

Враховуючи зазначені дидактичні принципи та необхідні функціональні можливості систем дистанційного навчання, можемо синтезувати такі модулі системи:

- модуль адміністрування системи;
- модуль організації та підтримки навчального процесу;
- модуль розроблення та підтримки тестів;
- модуль розроблення та представлення всіх видів навчальних матеріалів у системі;

- модуль експорту-імпорту навчальних матеріалів різноманітних форматів;
- модуль інтерактивної взаємодії користувачів курсів: лектор–студенти, студенти–студенти, студенти–лектор;
- модуль реєстру активності користувачів.

## 2.2. Огляд тренажерів зі схожою тематикою

Тренажерів, що закріплюють знання з алгебри предикатів та схожої тематики, не було знайдено. Але було розглянуто декілька з інших дисциплін.

Так тренажер з теми «М-метод для задач лінійного програмування» дистанційного навчального курсу «Методи оптимізації та дослідження операцій» імітує процес розв’язування задачі лінійного програмування М-методом на основі заданого прикладу [4, 5].

На стартовій сторінці виводиться тема програми і пропонується розпочати тренажер (рис. 2.1).

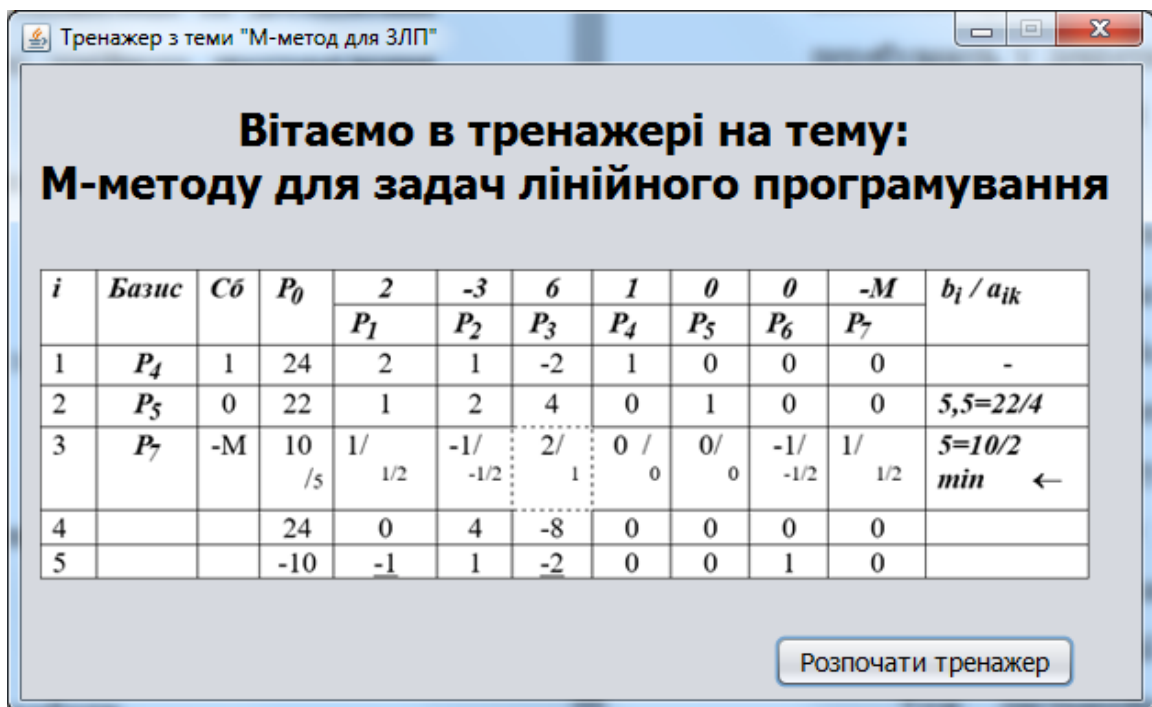


Рисунок 2.1 – Стартова сторінка тренажера з теми «М-метод для ЗЛП»

Один з варіантів завдання – це вибір правильної відповіді (рис. 2.2).

Тренажер з теми "М-метод для ЗЛП"

Чи можливо задану задачу розв'язувати М-методом?

$$F = -2x_1 + 3x_2 - x_3 - x_4 \rightarrow \min;$$
$$2x_1 + x_2 - 2x_3 + x_4 = 24;$$
$$x_1 + 2x_2 + x_3 \leq 22;$$
$$x_1 - x_2 + 2x_3 \geq 10;$$

Виберіть правильну відповідь:

☒ Так

☐ Ні

Наступне питання

Рисунок 2.2 – Завдання з вибором відповіді в тренажері з теми «М-метод для ЗЛП»

При неправильній відповіді виводиться помилка і пропонується перейти до наступного кроку (рис. 2.3).

Тренажер з теми "М-метод для ЗЛП"

Відповідь не вірна, дану задачу неможна розв'язати М-методом

Наступний крок

Рисунок 2.3 – Помилка в тренажері з теми «М-метод для ЗЛП»

Другий варіант завдання – це введення числа у активне поле, яких може бути декілька (рис. 2.4, 2.5).

Тренажер з теми "М-метод для ЗЛП"

Скільки умов повинно виконуватися, щоб задача розв'язувалася М-методом?

Введіть число:

Наступне питання

Рисунок 2.4 – Завдання з введенням числа в одне поле в тренажері з теми «М-метод для ЗЛП»

Тренажер з теми "М-метод для ЗЛП"

Який вигляд матиме цільова функція в канонічній формі?

$F =$    $2x_1$    $3x_2$    $x_3$    $x_4 \rightarrow$

Умова      Наступне питання

Рисунок 2.5 – Завдання з введенням числа в декілька полів в тренажері з теми «М-метод для ЗЛП»

Третій варіант завдання – заповнення таблиці. Схоже з попереднім варіантом, але є відмінність: перехід до наступного питання буде не активний доти, поки не заповнено всі комірки (рис. 2.6).

i	Базис	Сб	P <sub>0</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	b/a <sub>k</sub>
1	P <sub>4</sub>	1									
2	P <sub>5</sub>	0									
3	P <sub>7</sub>	-M									
4											
5											

Рисунок 2.6 – Завдання із заповненням таблиці в тренажері з теми «М-метод для ЗЛП»

При цьому є можливість переглянути умову задачі (рис. 2.7).

Цільова функція:  
 $F = 2x_1 - 3x_2 + 6x_3 + x_4 - Mx_7 \rightarrow \max;$

Умови:  
 $2x_1 + x_2 - 2x_3 + x_4 = 24;$   
 $x_1 + 2x_2 + 4x_3 + x_5 = 22;$   
 $x_1 - x_2 + 2x_3 - x_6 + x_7 = 10;$   
 $x_j \geq 0, j = 1..7;$

Рисунок 2.7 – Умова задачі в тренажері з теми «М-метод для ЗЛП»



Після проходження тренажера виводиться кількість допущених помилок (рис. 2.8).

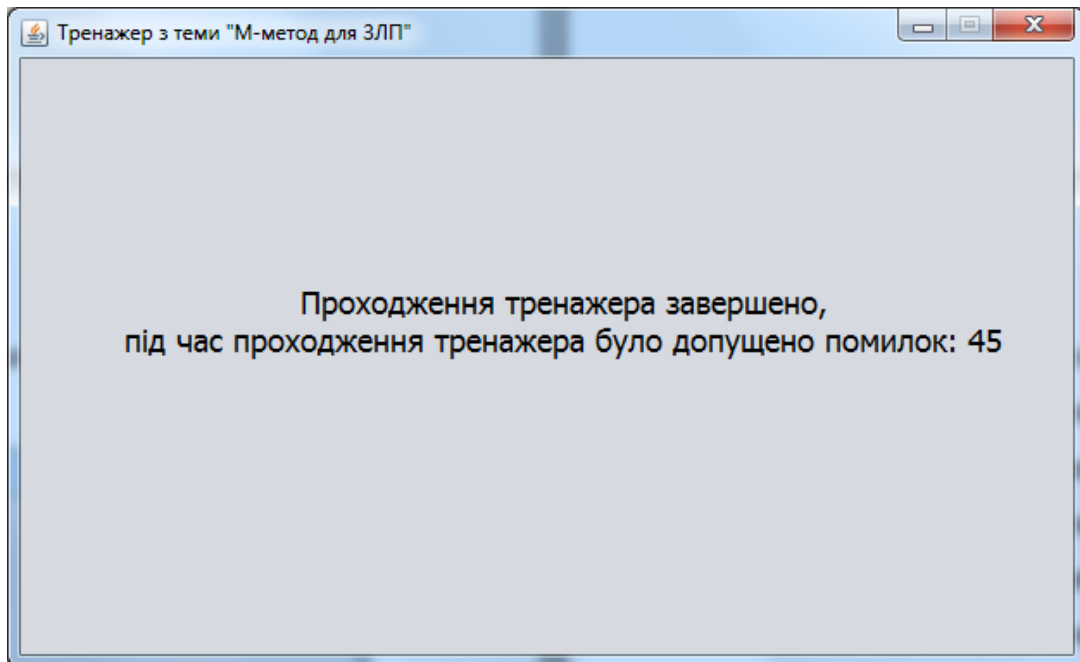


Рисунок 2.8 – Умова задачі в тренажері з теми «М-метод для ЗЛП»

Тренажер з теми «Системи числення» дистанційного навчального курсу «Інформаційні мережі» навчає студентів системам числення і основним операціям в них (рис. 2.9) [6, 7].



Рисунок 2.9 – Стартова сторінка тренажера з теми «Системи числення»

Пропонується перейти до теоретичного матеріалу або до операцій в системах числення (рис. 2.10, 2.11).

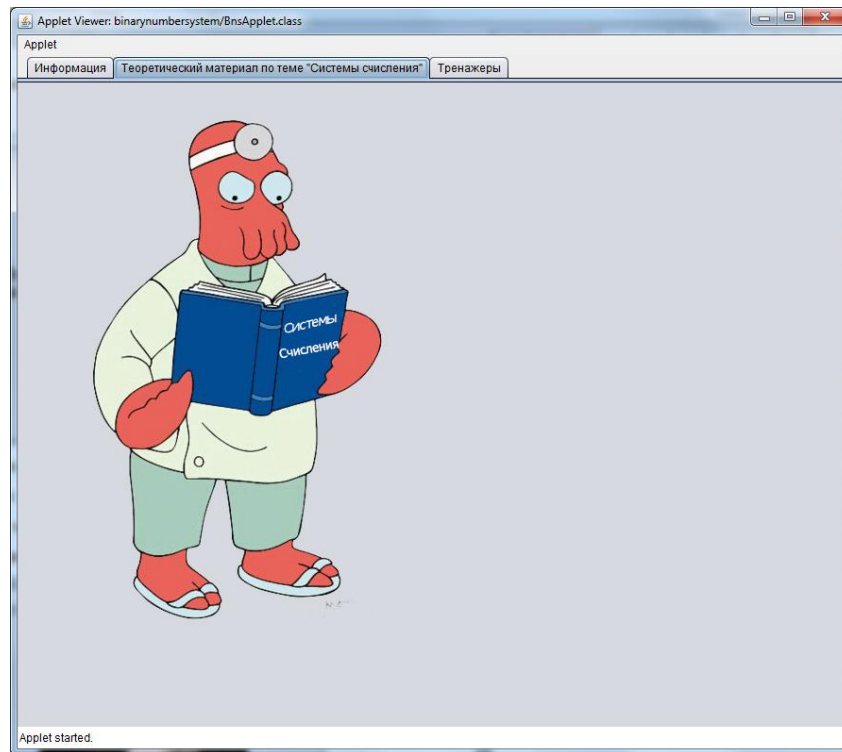


Рисунок 2.10 – Теоретичний матеріал в тренажері з теми «Системи числення»

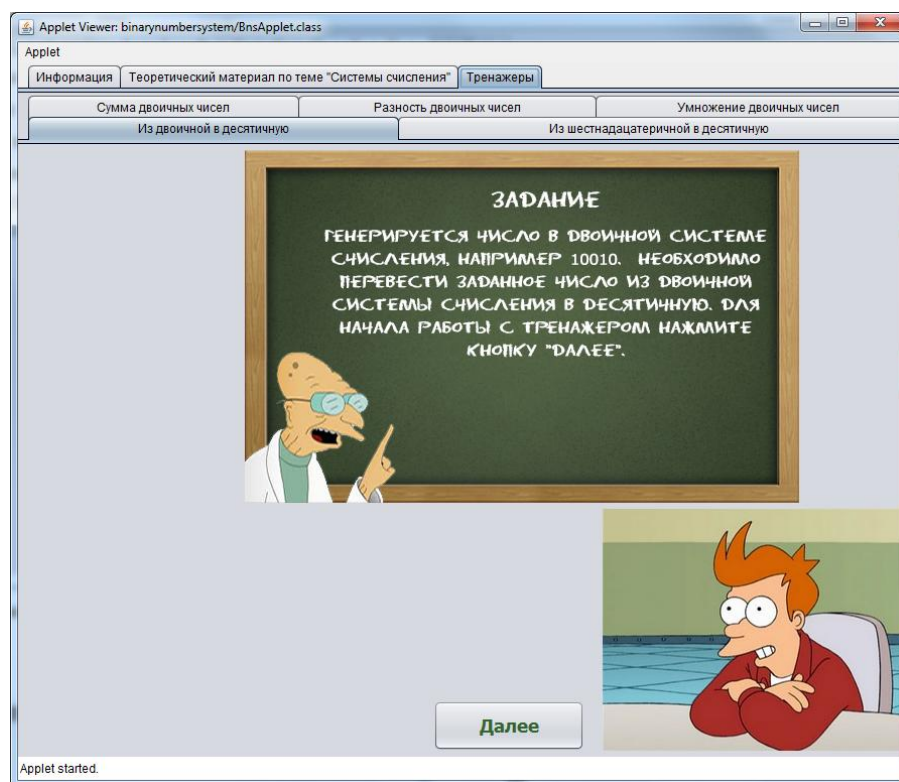


Рисунок 2.11 – Основні операції в тренажері з теми «Системи числення»

По кожній операції наводиться декілька завдань (рис. 2.12).

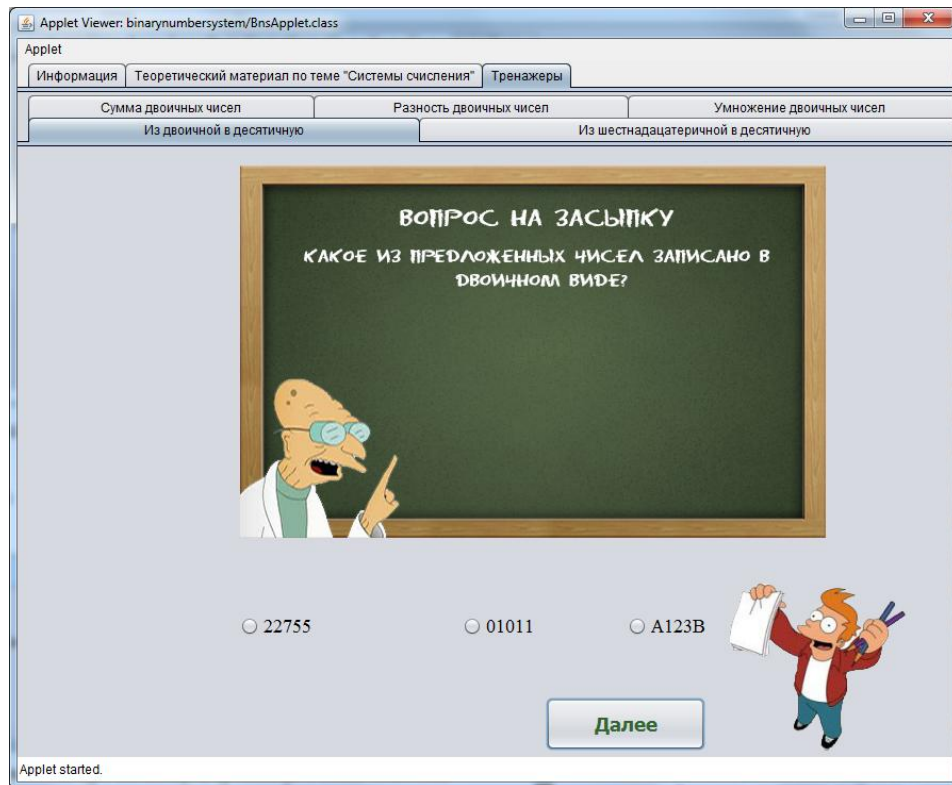


Рисунок 2.12 – Завдання з вибором відповіді в тренажері з теми «Системи числення»

Якщо відповісти неправильно, то виведеться повідомлення про помилку (рис. 2.13).

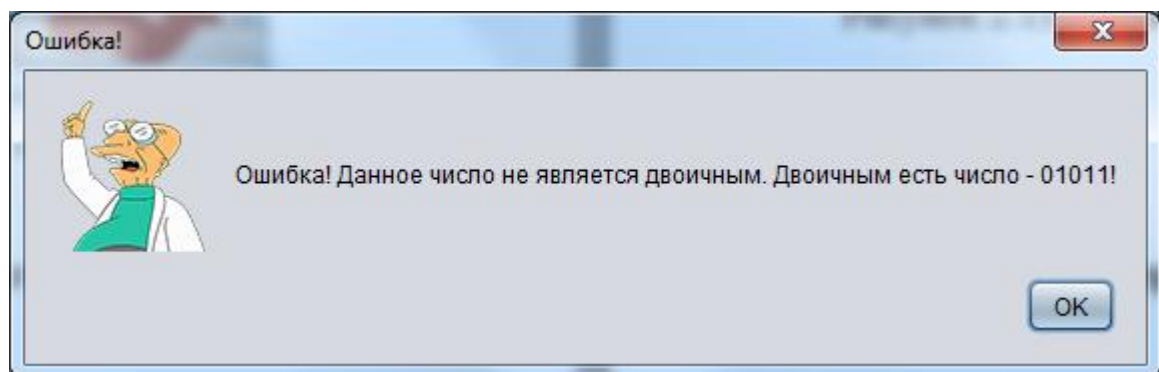


Рисунок 2.13 – Повідомлення про помилку в тренажері з теми «Системи числення»

Крім завдань з вибором також є заповнення активних комірок (рис. 2.14).

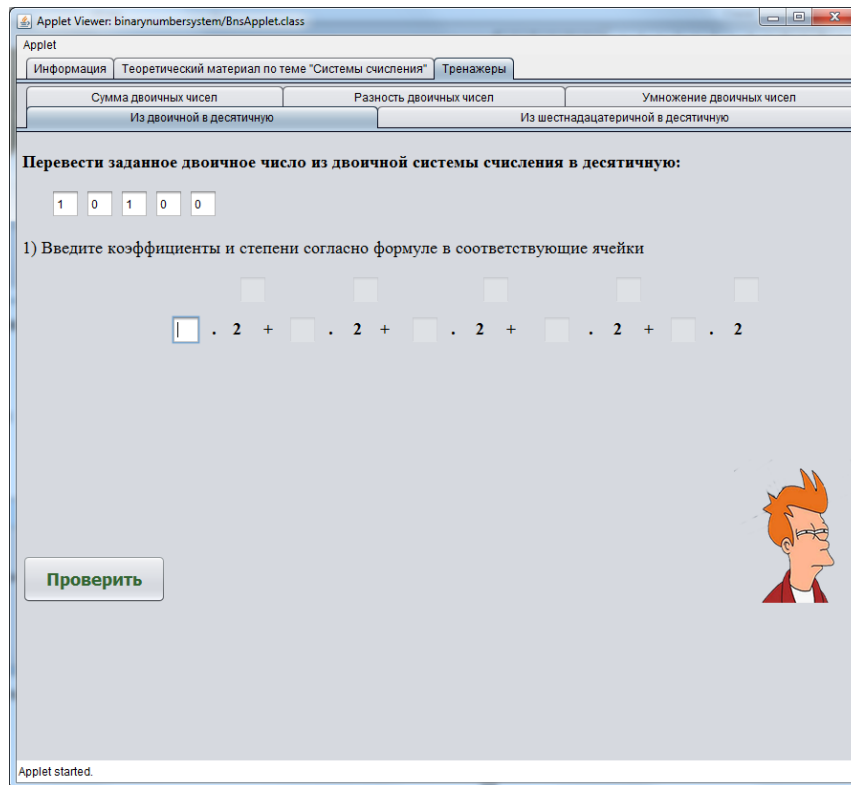


Рисунок 2.14 – Завдання із заповненням комірок в тренажері з теми «Системи числення»

### 2.3. Переваги та вади розглянутих тренажерів

До переваг розглянутих тренажерів можна віднести наступне:

- Наявність основної інформації про тренажер на стартовій сторінці;
- Наявність теоретичного матеріалу і його доступність;
- Красивий дизайн;
- Перевірка вибраної відповіді;
- Виведення помилки у разі неправильної відповіді;
- Декілька прикладів для закріплення знань з теми.

До вад можна віднести наступне:

- Відсутність теоретичного матеріалу;
- Занадто великі розміри тренажера;
- Після проходження надається можливість лише завершити роботу.

## **2.4. Актуальність теми роботи**

Досить часто дистанційне навчання плутають із заочною освітою, але це не одне і теж. Якщо, навчаючись на заочному, студент зустрічається з викладачем на вичитування лекцій та іспитах, а основний матеріал освоює самостійно і до всього доходить сам, то дистанційне навчання має на увазі, що обидві сторони знаходяться на зв'язку практично постійно і спілкуються за допомогою передових технологій.

Головною метою створення системи дистанційної освіти є забезпечення загальнонаціонального доступу до освітніх ресурсів шляхом використання сучасних інформаційних технологій та телекомунікаційних мереж і створення умов для реалізації громадянами своїх прав на освіту.

Дистанційне навчання забезпечує можливість навчатися тоді, коли зручно студенту, у тому темпі, що він сам обирає (в рамках установлених строків проведення курсів), в тому місці де він перебуває (не має потреби витрачати час на дорогу до ВНЗ, для здачі поточних, а іноді і підсумкових контролів). Тому до контингенту потенційних студентів дистанційної форми навчання можна віднести тих, хто часто перебуває у відрядженнях, військовослужбовців, територіально віддалених слухачів, жінок, що перебувають у декретній відпустці, людей з фізичними вадами, тих, хто поєднує навчання й роботу, співробітників, що підвищують свою кваліфікацію.

Варто зазначити, що дистанційне навчання в сучасному світі здійснюється за допомогою різноманітних технологій, які відрізняються за:

формою подання учбових матеріалів; наявністю посередника в системі навчання або за централізованою формою навчання; за ступенем використання телекомунікацій і персональних комп'ютерів; за технологією організації контролю учбового процесу; за ступенем впровадження в технології навчання звичайних методів ведення освітнього процесу; за методами ідентифікації студентів при складанні іспитів.

Тож дистанційні технології навчання можна розглядати як природний етап еволюції традиційної системи освіти від дошки з крейдою до електронної дошки й комп'ютерних навчальних систем, від книжкової бібліотеки до електронної, від звичайної аудиторії до віртуальної аудиторії.

Вивчення досвіду впровадження дистанційної форми навчання в системи освіти різних країн дозволило виділити основні мотиваційні причини, що спонукають навчальні заклади до організації системи дистанційного навчання: поліпшення якості навчання; переваги нових педагогічних технологій; зростаючий попит на нову форму навчання; отримання доходів; можливість постійної взаємодії викладачів і студентів; необхідність виживання в інформаційному суспільстві; можливість скорочення витрат на реорганізацію освіти.

Предикат - в традиційній логіці один з двох термінів думки, а саме той, в якому щось мовиться про предмет мови (суб'єкт). До кінця 19 ст. у логіці суб'єкт думки, як правило, ототожнювався з граматичним підметом, а предикат - з іменною частиною граматичного присудка, що виражається, наприклад, прикметником, традиційним оглядом. Форма присудка (предикативний зв'язок) зводилася до атрибутивного зв'язку, означала, що предмету (суб'єктові) властива певна ознака.

Новий погляд характеризується узагальненням поняття "Предиката" на основі поняття особливого роду функції - логічної (або пропозиційної) функції, значеннями якої служать вислови (або їх істинне значення - "істина" і "брехня").

В математичній логіці функції, значеннями яких служать вислови (або їх істинне значення "відмітка" і "брехня"), і називають предикат. Новий погляд на логічну структуру думки зводиться до того, що традиційні поняття предиката і суб'єкта замінюються відповідно на точні математичні поняття функції і її аргументів.

Відповідно до цього предикат визначаються на множинах (областях предметів), елементи яких служать аргументами, або значеннями відповідних змінних, нове трактування предикат додає необхідну спільність логічному міркуванню, яке об'єднує висновки не тільки силогізму, але і несилогізму, а функціональна форма запису відкриває широкі можливості для формалізації висловів будь-якій науковій теорії.

### 3. ТЕОРЕТИЧНА ЧАСТИНА

#### 3.1. Алгебра предикатів

Логіка предикатів — це розділ класичної символічної логіки, що вивчає суб'єктно-предикатну структуру висловлювань, на підставі чого визначають значення істинності висловлювань; по-іншому — це дедуктивна теорія, яка моделює процес виведення одних висловлювань із інших, враховуючи їх структуру. Логіку предикатів трактують як розширення логіки висловлювань через виявлення внутрішньої структури висловлювань і введення нових термінів та системи аксіом.

Логіка предикатів як система створюється відповідно до загальних принципів побудови формальних систем. Особливість логіки предикатів полягає в тому, що вона є складнішою і за семантикою, і за синтаксисом порівняно з логікою висловлювань. Розрізняють семантику та синтаксис логіки предикатів.

У семантичному аспекті визначають суб'єктно-предикатну структуру висловлювань на змістовному рівні. Це дає змогу виявити властивості, притаманні певній сукупності емпіричних або абстрактних об'єктів, і ввести терміни, котрі відокремлюють сферу дії предикатів: висловлювання, властивість, відношення, предикат, одномісний предикат, багатомісний предикат, квантор загальності, квантор існування, істинне значення висловлення.

Висловлення, в якому емпіричному чи абстрактному об'єктові приписують певну властивість  $P$  або визначаються відношення між об'єктами, надають два значення істинності: «істина» ( $i$ ); «хиба» ( $x$ ). Відповідно, логіка предикатів — двозначна за кількістю значень істинності висловлювань.



У синтаксичному аспекті суб'єктно-предикатну структуру висловлювань визначають у процесі абстрагування від їх змісту та формалізують засобами штучно створеної мови, на підставі чого здійснюють логічні операції над символами, що зображають ці відношення (числення предикатів).

Структура логіки предикатів — алфавіт, правила побудови формул із символів алфавіту, правила дедуктивного виведення з аксіом нових формул (доведення теорем), правила інтерпретації.

Мова логіки предикатів — це система символів, що створюють алфавіт. До нього належать символи, введені в логіці висловлювань, і нові символи, які позначають терміни, введені в логіці предикатів.

Алфавіт:

- маленькі латинські літери можливо з індексами або без них, які називаються предикатними змінними або термами;
- великі латинські літери з індексами знизу або без них, які називаються висловлюваними змінними;
- $P_n(x_1, \dots, x_n)$ ,  $Q_n(x_1, \dots, x_n)$  предикатні змінні;
- символи логічних операцій  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$ ;
- символи кванторів  $\exists$ ,  $\forall$ ;
- технічні символи: ( — ліва дужка;) — права дужка.

Предикат ( $n$ -місний, або  $n$ -арний) — це функція з областю значень  $\{0,1\}$  (або «Істина» та «Хибна»), певна на  $n$ -й декартовій ступені множини  $M$ . Таким чином, кожному  $n$ -ку елементів  $M$  він характеризує або як «справжню», або як «неправдиву».

Під  $n$ -місним предикатом  $P(x_1, \dots, x_n)$  будемо розуміти деяку логічну функцію  $n$  змінних  $x_1, \dots, x_n$ , що визначена на множині  $\Omega$  і приймає значення істина або хибне.

Область визначення предиката — множина  $\Omega$  на якій визначений предикат.

Квантор загальності позначає висловлювання, в якому властивість  $P$  приписують певному непорожньому класу загалом, що означає: для всіх елементів класу  $A$  притаманна властивість  $P$ . Цей квантор має вираз «для всіх» («усі», «кожний», «будь-який», «який би не був»). Його позначають символом  $\forall$ , а повна формула —  $\forall xP(x)$  (чит. кожному  $x$  притаманна властивість  $P$ ). Так, висловлювання "Для всіх індивідів класу людей притаманна властивість «бути смертними»" («Усі люди смертні») зображають формулою  $\forall xP(x)$ .

Квантор існування позначає висловлювання про певний непорожній клас, в якому властивість  $P$  притаманна лише деяким елементам цього класу, тобто існують елементи класу  $A$ , яким притаманна властивість  $P$ .

Квантор існування має вираз «існує» («деякі», «лише один»). Його позначають символом  $\exists$ , а повна формула —  $\exists xP(x)$  (чит. «існує»  $x$ , яке має властивість  $P$ ). Наприклад, висловлювання «Існують люди, котрим притаманна властивість писати вірші» («Деякі люди пишуть вірші») зображають формулою  $\exists xP(x)$ .

Квантори загальності й існування взаємозалежні, тому всі логічні операції здійснюють з визначенням логічних відношень над ними.

Побудова формул логіки предикатів:

- Окремо взятий предикат називається елементарною формулою.
- Якщо  $F$  і  $Q$  — формули логіки предикатів, то  $\neg F$ ,  $(F \wedge Q)$ ,  $(F \vee Q)$ ,  $(F \rightarrow Q)$  — формули.
- $P(x)$  — формула, що виражає властивість (одномісний предикат).
- $R(x, y)$  — формула, яка виражає двомісний предикат.
- $R(x, y, z)$  — формула, що виражає тримісний предикат.
- Якщо  $P$  — формула і  $x$  — предметна змінна, то  $\forall x P(x)$  і  $\exists x P(x)$  є формулами.

Область дії квантора означає вираз, до якого належить квантор. ОДК обмежують дужками зліва і справа від виразу. Ліва дужка означає початок сфери дії, а права дужка — закінчення. У межах ОДК виокремлюють зв'язану та вільну змінні. Змінну, що слідує безпосередньо після квантора, називають підкванторною змінною, а формула, до якої належить квантор, — підкванторною формулою, або сферою дії квантора. Зв'язана змінна — змінна, яка входить до сфери дії кванторів загальності  $\forall$  чи існування  $\exists$  або обох відразу. Наприклад, у формулах  $\forall xP(x)$ ,  $\exists xP(x)$  зв'язаною змінною є  $x$ .

Вільна змінна входить до певної формули, але не входить до сфери дії кванторів загальності  $\forall$  чи існування  $\exists$  на відміну від зв'язаної змінної. Так, у формулі  $\neg(P(x)) \rightarrow Q(x)$  — змінна  $x$  зв'язана так само, як у формулі  $\forall xP(x)$ , але вільна у виразі  $Q(x)$ .

У логіці предикатів квантор загальності трактують як узагальнення кон'юнкції, а квантор існування — як узагальнення диз'юнкції, якщо множинність  $M$  значень змінної  $x$  є скінченною, тобто вона складається зі скінченної кількості предметів.

Квантифікація — визначення обсягу суб'єкта та предиката в структурі висловлювання за допомогою кванторних термінів — «усі» («будь-який», «кожний») та «деякі»; логічна операція, за допомогою якої визначають сферу дії кванторів. Це перехід від формули виду  $P(x)$  до формули виду  $\exists x(P(x))$  або  $\forall x(P(x))$ , унаслідок чого змінна  $x$  у формулі  $P(x)$  перестає бути просто символом, а виражає певну властивість, притаманну класові  $A$ . Змінну  $x$  у формулі  $P(x)$  називають вільною змінною, а після квантифікації — зв'язаною змінною, тобто у формулах  $\exists x(P(x))$  і  $\forall x(P(x))$  змінна  $x$  стає зв'язаною. Квантифікація висловлювань набувають такого вигляду:  $P(x, y)$  — двомісний предикат, визначений на множинності  $M$ . Квантор загальності та квантор існування можна використати і для змінної

$x$  і для змінної  $y$ . Змінна, до якої використано квантор, стає зв'язаною, а друга змінна — вільною.

За допомогою квантифікації (використання квантора для однієї зі змінних) двомісний предикат можна перетворити на одномісний, а тримісний — в двомісний. Значення істинності висловлювань з кванторами загальності й існування. Логіка предикатів є двозначною за кількістю значень істинності, тому висловлюванням із кванторами загальності й існування надають два значення істинності — «і», «х». Для визначення істинності висловлювання з кванторами загальності або існування задають множину  $M$  з певною кількістю елементів, для якої предикат є істинним. Значення істинності визначають за допомогою таблиці істинності.

Логіка першого порядку (числення предикатів) — це формальна система в математичній логіці, в якій допускаються висловлення відносно змінних, фіксованих функцій, і предикатів. Є розширенням логіки висловлювань. В свою чергу є частковим випадком логіки вищого порядку

Мови логіки першого порядку будуються на основі: множини функціональних символів  $F$  і множини предикатних символів  $P$ . З кожним функціональним і предикатним символом пов'язана арність (число аргументів). Крім того використовуються додаткові символи:

- Символи змінних;
- Пропозиційні зв'язки:  $\vee$ ,  $\wedge$ ,  $\neg$ ,  $\rightarrow$ ;
- Квантори: загальності  $\forall$  та існування  $\exists$ ;
- Службові символи: дужки і кома.

Перелічені символи разом із символами з  $P$  і  $F$  утворюють Алфавіт логіки першого порядку [8, 9].

### 3.2. Приклади застосування алгебри предикатів

*Приклад 1.* Навести приклади інтерпретацій формули алгебри предикатів  $\forall x(P(x) \rightarrow Q(x))$ .

*Розв'язання.*

1) Область інтерпретації – множина живих істот,  $P(x)$  :  $x$  – риба,  $Q(x)$  :  $x$  живе у воді. Інтерпретація формули: «Всі риби живуть у воді».

2) Область інтерпретації – множина живих істот,  $P(x)$  :  $x$  – людина,  $Q(x)$  :  $x$  смертний. Інтерпретація формули: «Всі люди смертні».

3) Область інтерпретації – множина цілих чисел,  $P(x)$  :  $x$  ділиться на 6,  $Q(x)$  :  $x$  ділиться на 3. Інтерпретація формули: «Всі числа, які діляться на 6, діляться на 3».

*Приклад 2.*

Нехай предикат  $P(x)$  відповідає реченню « $x \geq 1$ », а предметна область складається з усіх дійсних чисел. Тоді висловлювання  $\forall x P(x)$  хибне:

$$\forall x P(x) = F.$$

Якщо ж предмета область складається з усіх натуральних чисел, то висловлювання  $\forall x P(x)$  істинне:

$$\forall x P(x) = T.$$

*Приклад 3.*

Позначимо речення « $x$  — просте число» як  $P(x)$ , « $x$  — раціональне число» — як  $Q(x)$ , « $x$  — дійсне число» як  $R(x)$  та « $x$  менше  $y$ » — як  $МЕНШЕ(x, y)$ . Розглянемо такі істинні твердження.

1. Кожне раціональне число дійсне.
2. Існує просте число.
3. Для кожного числа  $x$  існує таке число  $y$ , що  $x < y$ .

Наведені речення можна записати такими формулами.

1.  $\forall x(Q(x) \rightarrow R(x))$ .

2.  $\exists x P(x)$ .

3.  $\forall x \exists y \text{ММЕНШ}(x, y)$ .

*Приклад 4.*

Запишемо речення «Кожний студент групи вивчав дискретну математику» за допомогою предикатів і кванторів.

Спочатку перепишемо речення так, щоб було зрозуміло, як краще розставити квантори: «Про кожного студента групи відомо, що цей студент вивчав дискретну математик».

Тепер уведемо змінну  $x$ , і речення набере вигляду: «Про кожного студента  $x$  групи відомо, що  $x$  вивчав дискретну математику».

Уведемо предикат  $C(x)$ : « $x$  вивчав дискретну математику». Якщо предметна область змінної  $x$  — усі студенти групи, то можна записати задане речення як  $\forall x C(x)$ .

Є й інші коректні подання з різними предметними областями та предикатами. Зокрема, можна вважати, що нас цікавлять інші групи людей, окрім тих, які вчаться в одній академічній групі.

Узявши як предметну область усіх людей, можна записати задане речення так: «Для кожної особи  $x$ , якщо ця особа  $x$  — студент групи, то  $x$  вивчав дискретну математику».

Якщо предикат  $S(x)$  має вигляд «Особа  $x$  учиться в групі», то задане речення треба записати у вигляді  $\forall x (S(x) \rightarrow C(x))$ . Зауважимо, що задане речення не можна записати як  $\forall x (S(x) \wedge C(x))$ , бо тоді це означало б, що всі особи з предметної області вчаться в групі та вивчали дискретну математику.

Іще один спосіб записати задане речення — це ввести двомісний предикат  $Q(x, y)$ : «Студент  $x$  вивчає дисципліну  $y$ ». Тоді можна замінити  $C(x)$  на  $Q(x, \text{Дискретна\_математика})$ , що дасть можливість переписати наведені формули у вигляді  $\forall x Q(x, \text{Дискретна\_математика})$  чи  $\forall x (S(x) \rightarrow Q(x, \text{Дискретна\_математика}))$  [9, 10].

### 3.3. Алгоритмізація за темою роботи

Головна сторінка тренажера містить інформацію щодо назви тренінгу, його розробника та керівника. Також користувачу надається можливість переглянути теоретичний матеріал і приклади, перейти до проходження тренінгу. На кожному кроці виводиться умова прикладу, завдання та наводяться варіанти відповіді. Лише один з варіантів відповідей є вірним.

**Крок 1.** Користувачу виводиться умова і завдання: «Нехай предикат  $P(x)$  відповідає реченню « $x \geq 1$ ». Вказати предметну область, при якій висловлювання  $\forall x P(x)$  істинне». Наводяться варіанти відповіді:

- Предметна область – множина дійсних чисел;
- Предметна область – множина натуральних чисел;
- Предметна область – множина цілих чисел;
- Всі відповіді вірні.

Якщо користувач обирає правильну відповідь, тобто другий варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: «Якщо предмета область складається з усіх натуральних чисел, то висловлювання  $\forall x P(x)$  істинне».

**Крок 2.** Користувачу виводиться умова і завдання: «Нехай предикат  $P(x)$  відповідає реченню « $x \geq 1$ ». Вказати предметну область, при якій висловлювання  $\exists x P(x)$  істинне». Наводяться варіанти відповіді:

- Предметна область – множина дійсних чисел;
- Предметна область – множина натуральних чисел;
- Предметна область – множина цілих чисел;
- Всі відповіді вірні.

Якщо користувач обирає правильну відповідь, тобто четвертий варіант, то відбувається перехід на крок 3, інакше – виводиться повідомлення про помилку: «Висловлювання  $\exists x P(x)$  завжди істинне, якщо

предмета область складається або з усіх натуральних чисел, або цілих чисел, або дійсних чисел».

**Крок 3.** Користувачу виводиться умова і завдання: «Нехай предикат  $P(x)$  відповідає реченню « $x$  – просте число» та задано істинне твердження «Існує просте число». Якою формулою можна записати вказане речення?». Наводяться варіанти відповіді:

- $\exists x P(x)$ ;
- $\forall x P(x)$ ;
- $\forall x \exists x P(x)$ ;
- $P(x)$ .

Якщо користувач обирає правильну відповідь, тобто перший варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: «Твердження «Існує просте число» при  $P(x)$  : « $x$  – просте число» можна записати формулою  $\exists x P(x)$ ».

**Крок 4.** Користувачу виводиться умова і завдання: «Нехай предикат  $Q(x)$  відповідає реченню « $x$  – раціональне число»,  $R(x)$  відповідає реченню « $x$  – дійсне число» та задано істинне твердження «Кожне раціональне число дійсне». Якою формулою можна записати вказане речення?». Наводяться варіанти відповіді:

- $\exists x (Q(x) \rightarrow R(x))$ ;
- $\forall x \exists x (Q(x) \rightarrow R(x))$ ;
- $\forall x (Q(x) \rightarrow R(x))$ ;
- $Q(x) \rightarrow R(x)$ .

Якщо користувач обирає правильну відповідь, тобто третій варіант, то відбувається перехід на крок 5, інакше – виводиться повідомлення про помилку: «Твердження «Кожне раціональне число дійсне» при  $Q(x)$  : « $x$  – раціональне число»,  $R(x)$  : « $x$  – дійсне число» можна записати формулою  $\forall x (Q(x) \rightarrow R(x))$ ».



**Крок 5.** Користувачу виводиться умова і завдання: «Нехай предикат  $P(x, y)$  відповідає реченню « $x$  менше  $y$ » та задано істинне твердження «Для кожного числа  $x$  існує таке число  $y$ , що  $x < y$ ». Якою формулою можна записати вказане речення?». Наводяться варіанти відповіді:

- $\exists y P(x, y)$ ;
- $\forall x P(x, y)$ ;
- $\forall x \exists y P(x, y)$ ;
- $P(x, y)$ .

Якщо користувач обирає правильну відповідь, тобто третій варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: «Твердження «Для кожного числа  $x$  існує таке число  $y$ , що  $x < y$ » при  $P(x)$  : « $x$  менше  $y$ » можна записати формулою  $\forall x \exists y P(x, y)$ ».

**Крок 6.** Користувачу виводиться умова і завдання: «Навести інтерпретацію формули алгебри предикатів  $\forall x(P(x) \rightarrow Q(x))$ , де область інтерпретації – множина живих істот,  $P(x)$  :  $x$  – риба,  $Q(x)$  :  $x$  живе у воді». Наводяться варіанти відповіді:

- Всі риби живуть у воді;
- Існують риби, що живуть у воді;
- Не всі риби живуть у воді;
- Всі риби можуть жити у воді.

Якщо користувач обирає правильну відповідь, тобто перший варіант, то відбувається перехід на крок 7, інакше – виводиться повідомлення про помилку: «Інтерпретація формули  $\forall x(P(x) \rightarrow Q(x))$ : «Всі риби живуть у воді», де  $P(x)$  :  $x$  – риба,  $Q(x)$  :  $x$  живе у воді».

**Крок 7.** Користувачу виводиться умова і завдання: «Навести інтерпретацію формули алгебри предикатів  $\exists x(P(x) \rightarrow Q(x))$ , де область

інтерпретації – множина живих істот,  $P(x) : x$  – риба,  $Q(x) : x$  може жити без води». Наводяться варіанти відповіді:

- Всі риби можуть жити без води;
- Існують риби, що можуть жити без води;
- Всі риби не можуть жити без води;
- Всі риби можуть жити у воді.

Якщо користувач обирає правильну відповідь, тобто другий варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: «Інтерпретація формули  $\exists x(P(x) \rightarrow Q(x))$  : «Існують риби, що можуть жити без води», де  $P(x) : x$  – риба,  $Q(x) : x$  може жити без води».

**Крок 8.** Користувачу виводиться умова і завдання: «Навести інтерпретацію формули алгебри предикатів  $\forall x(P(x) \rightarrow Q(x))$ , де область інтерпретації – множина живих істот,  $P(x) : x$  – людина,  $Q(x) : x$  смертний». Наводяться варіанти відповіді:

- Всі люди безсмертні;
- Існують смертні люди;
- Не всі люди смертні;
- Всі люди смертні.

Якщо користувач обирає правильну відповідь, тобто четвертий варіант, то відбувається перехід на крок 9, інакше – виводиться повідомлення про помилку: «Інтерпретація формули  $\forall x(P(x) \rightarrow Q(x))$  : «Всі люди смертні», де  $P(x) : x$  – людина,  $Q(x) : x$  смертний».

**Крок 9.** Користувачу виводиться умова і завдання: «Навести інтерпретацію формули алгебри предикатів  $\forall x(P(x) \rightarrow Q(x))$ , де область інтерпретації – множина цілих чисел,  $P(x) : x$  ділиться на 6,  $Q(x) : x$  ділиться на 3». Наводяться варіанти відповіді:

- Всі числа, які діляться на 6, діляться на 3;
- Існують числа, які діляться на 3, якщо діляться на 6;

- Всі числа, які діляться на 3, діляться на 6;
- Не всі числа, які діляться на 6, діляться на 3.

Якщо користувач обирає правильну відповідь, тобто перший варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: «Інтерпретація формули  $\forall x(P(x) \rightarrow Q(x))$  : «Всі числа, які діляться на 6, діляться на 3», де  $P(x)$  :  $x$  ділиться на 6,  $Q(x)$  :  $x$  ділиться на 3, область інтерпретації – множина цілих чисел».

**Крок 10.** Користувачу виводиться умова і завдання: «Навести інтерпретацію формули алгебри предикатів  $\exists x(P(x) \rightarrow Q(x))$ , де область інтерпретації – множина цілих чисел,  $P(x)$  :  $x$  ділиться на 6,  $Q(x)$  :  $x$  ділиться на 3». Наводяться варіанти відповіді:

- Всі числа, які діляться на 6, діляться на 3;
- Існують числа, які діляться на 3, якщо діляться на 6;
- Всі числа, які діляться на 3, діляться на 6;
- Існують числа, які діляться на 6, якщо діляться на 3.

Якщо користувач обирає правильну відповідь, тобто другий варіант, то відбувається перехід на крок 11, інакше – виводиться повідомлення про помилку: «Інтерпретація формули  $\exists x(P(x) \rightarrow Q(x))$  : «Існують числа, які діляться на 3, якщо діляться на 6», де  $P(x)$  :  $x$  ділиться на 6,  $Q(x)$  :  $x$  ділиться на 3, область інтерпретації – множина цілих чисел».

**Крок 11.** Користувачу виводиться умова і завдання: «Записати речення «Кожний студент групи вивчав дискретну математику» за допомогою предикатів і кванторів. Спочатку переписати речення так, щоб було зрозуміло, як краще розставити квантори». Наводяться варіанти відповіді:

- Про кожного студента відомо, що цей студент вивчав дискретну математику;
- Про кожного студента групи відомо, що цей студент вивчав;

- Про студента групи відомо, що цей студент вивчав дискретну математику;
- Про кожного студента групи відомо, що цей студент вивчав дискретну математику.

Якщо користувач обирає правильну відповідь, тобто четвертий варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: «Переписане речення має вигляд: «Про кожного студента групи відомо, що цей студент вивчав дискретну математику»».

**Крок 12.** Користувачу виводиться умова і завдання: «Перепишемо речення так, щоб було зрозуміло, як краще розставити квантори: «Про кожного студента групи відомо, що цей студент вивчав дискретну математик». Тепер уведемо змінну  $x$ , і речення набере вигляду: ». Наводяться варіанти відповіді:

- Про студента  $x$  групи відомо, що  $x$  вивчав дискретну математику;
- Про кожного студента  $x$  групи відомо, що  $x$  вивчав дискретну математику;
- Про кожного студента групи відомо, що  $x$  вивчав дискретну математику;
- Про кожного студента  $x$  групи відомо, що вивчав дискретну математику.

Якщо користувач обирає правильну відповідь, тобто другий варіант, то відбувається перехід на крок 13, інакше – виводиться повідомлення про помилку: «Речення набере вигляду: «Про кожного студента  $x$  групи відомо, що  $x$  вивчав дискретну математику»».

**Крок 13.** Користувачу виводиться умова і завдання: «Тепер уведемо змінну  $x$ , і речення набере вигляду: «Про кожного студента  $x$  групи відомо, що  $x$  вивчав дискретну математику». Уведемо предикат  $C(x)$ :». Наводяться варіанти відповіді:

- $x$  вивчав дискретну математику;
- $x$  – студент групи;
- студент вивчав  $x$ ;
- вивчав дискретну математику.

Якщо користувач обирає правильну відповідь, тобто перший варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: «Предикат  $C(x)$ : « $x$  вивчав дискретну математику»».

**Крок 14.** Користувачу виводиться умова і завдання: «Уведемо предикат  $C(x)$ : « $x$  вивчав дискретну математику». Якщо предметна область змінної  $x$  — усі студенти групи, то можна записати задане речення як». Наводяться варіанти відповіді:

- $\exists x C(x)$ ;
- $\forall x P(x)$ ;
- $\forall x C(x)$ ;
- $\forall x \exists x C(x)$ .

Якщо користувач обирає правильну відповідь, тобто третій варіант, то відбувається перехід на крок 15, інакше – виводиться повідомлення про помилку: «Якщо предметна область змінної  $x$  — усі студенти групи, то можна записати задане речення як  $\forall x C(x)$ ».

**Крок 15.** Користувачу виводиться умова і завдання: «Слід вважати, що нас цікавлять інші групи людей, окрім тих, які вчаться в одній академічній групі. Узявши як предметну область усіх людей, можна записати задане речення так:». Наводяться варіанти відповіді:

- Для кожної особи, якщо ця особа  $x$  — студент групи, то  $x$  вивчав дискретну математику;
- Для кожної особи  $x$ , якщо ця особа — студент групи, то  $x$  вивчав дискретну математику;

- Для кожної особи  $x$ , якщо ця особа  $x$  — студент групи, то  $x$  вивчав дискретну математику;
- Для кожної особи  $x$ , якщо ця особа  $x$  — студент групи, то вивчав дискретну математику.

Якщо користувач обирає правильну відповідь, тобто третій варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: «Узявши як предметну область усіх людей, можна записати задане речення так: «Для кожної особи  $x$ , якщо ця особа  $x$  — студент групи, то  $x$  вивчав дискретну математику»».

**Крок 16.** Користувачу виводиться умова і завдання: «Узявши як предметну область усіх людей, можна записати задане речення так: «Для кожної особи  $x$ , якщо ця особа  $x$  — студент групи, то  $x$  вивчав дискретну математику». Визначити предикат  $S(x)$ ». Наводяться варіанти відповіді:

- Особа  $x$  учиться в групі;
- Особа  $x$  вивчала дискретну математику;
- Особа  $y$  учиться в групі;
- Особа  $x$  учиться в групі.

Якщо користувач обирає правильну відповідь, тобто четвертий варіант, то відбувається перехід на крок 17, інакше – виводиться повідомлення про помилку: «Предикат  $S(x)$  має вигляд «Особа  $x$  учиться в групі»».

**Крок 17.** Користувачу виводиться умова і завдання: «Якщо предикат  $S(x)$  має вигляд «Особа  $x$  учиться в групі», то задане речення треба записати у вигляді». Наводяться варіанти відповіді:

- $\exists x(S(x) \rightarrow C(x))$ ;
- $\forall x(S(x) \rightarrow C(x))$ ;
- $\forall x \exists x(S(x) \rightarrow C(x))$ ;
- $\forall x(S(x) \wedge C(x))$ .

Якщо користувач обирає правильну відповідь, тобто другий варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: «Якщо предикат  $S(x)$  має вигляд «Особа  $x$  учиться в групі», то задане речення треба записати у вигляді  $\forall x(S(x) \rightarrow C(x))$ ».

**Крок 18.** Користувачу виводиться умова і завдання: «Якщо предикат  $S(x)$  має вигляд «Особа  $x$  учиться в групі», то задане речення треба записати у вигляді  $\forall x(S(x) \rightarrow C(x))$ . Зауважимо, що задане речення не можна записати як  $\forall x(S(x) \wedge C(x))$ , бо тоді б це означало:». Наводяться варіанти відповіді:

- Всі особи з предметної області вчаться в групі та вивчали дискретну математику;
- Існують особи з предметної області, що вчаться в групі та вивчали дискретну математику;
- Всі особи з предметної області вчаться в групі та не вивчали дискретну математику;
- Всі особи з предметної області не вчаться в групі, але вивчали дискретну математику.

Якщо користувач обирає правильну відповідь, тобто перший варіант, то відбувається перехід на наступний крок, інакше – виводиться повідомлення про помилку: « $\forall x(S(x) \wedge C(x))$  означало б «Всі особи з предметної області вчаться в групі та вивчали дискретну математику».

**Крок 19.** Користувачу виводиться умова і завдання: «Якщо ввести двомісний предикат  $Q(x, y)$ : «Студент  $x$  вивчає дисципліну  $y$ », то можна замінити». Наводяться варіанти відповіді:

- $S(x)$  на  $Q(x, \text{Дискретна\_математика})$ ;
- $C(x)$  на  $Q(y, \text{Дискретна\_математика})$ ;
- $C(x)$  на  $Q(\text{Дискретна\_математика}, x)$ ;
- $C(x)$  на  $Q(x, \text{Дискретна\_математика})$ .

Якщо користувач обирає правильну відповідь, тобто четвертий варіант, то відбувається перехід на крок 20, інакше – виводиться повідомлення про помилку: «Наведені формули можна переписати у вигляді  $\forall x Q(x, \text{Дискретна\_математика})$  чи  $\forall x (S(x) \rightarrow Q(x, \text{Дискретна\_математика}))$ ».

**Крок 20.** Користувачу виводиться умова і завдання: «Якщо ввести двомісний предикат  $Q(x, y)$ : «Студент  $x$  вивчає дисципліну  $y$ », то можна замінити  $C(x)$  на  $Q(x, \text{Дискретна\_математика})$ , що дасть можливість переписати формулу  $\forall x C(x)$  у вигляді:». Наводяться варіанти відповіді:

- $\forall x Q(x, \text{Дискретна\_математика})$  ;
- $\forall x (S(x) \rightarrow C(x, \text{Дискретна\_математика}))$  ;
- $\forall x (S(x) \rightarrow Q(x, \text{Дискретна\_математика}))$  ;
- $\forall x C(x, \text{Дискретна\_математика})$  .

Якщо користувач обирає правильну відповідь, тобто перший варіант, то відбувається перехід на крок 21, інакше – виводиться повідомлення про помилку: «Формулу  $\forall x C(x)$  можна переписати у вигляді  $\forall x Q(x, \text{Дискретна\_математика})$ ».

**Крок 21.** Користувачу виводиться умова і завдання: «Якщо ввести двомісний предикат  $Q(x, y)$ : «Студент  $x$  вивчає дисципліну  $y$ », то можна замінити  $C(x)$  на  $Q(x, \text{Дискретна\_математика})$ , що дасть можливість переписати формулу  $\forall x (S(x) \rightarrow C(x))$  у вигляді:». Наводяться варіанти відповіді:

- $\forall x Q(x, \text{Дискретна\_математика})$  ;
- $\forall x (S(x) \rightarrow C(x, \text{Дискретна\_математика}))$  ;
- $\forall x (S(x) \rightarrow Q(x, \text{Дискретна\_математика}))$  ;
- $\forall x C(x, \text{Дискретна\_математика})$  .

Якщо користувач обирає правильну відповідь, тобто третій варіант, то відбувається перехід на крок 22, інакше – виводиться повідомлення про



помилку: «Формулу  $\forall x(S(x) \rightarrow C(x))$  можна переписати у вигляді  $\forall x(S(x) \rightarrow Q(x, \text{Дискретна\_математика}))$ ».

**Крок 22.** Користувачу відображається повідомлення про завершення проходження тренажера. Надається можливість пройти тренінг спочатку або завершити його роботу.

### 3.4. Розробка блок-схеми

На рисунках 3.1-3.21 зображено блок-схему алгоритму роботи тренажера.

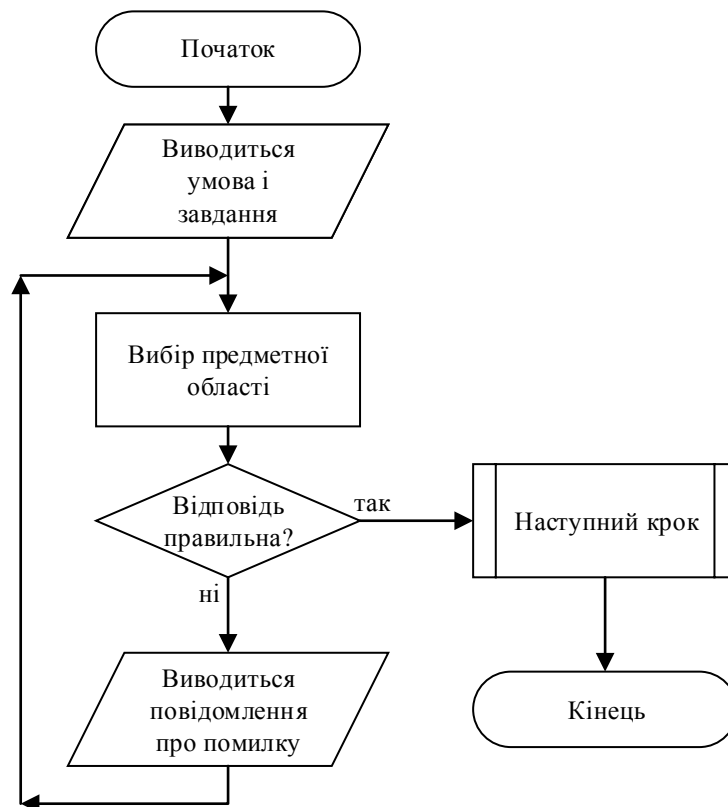


Рисунок 3.1 – Блок-схема першого кроку алгоритму

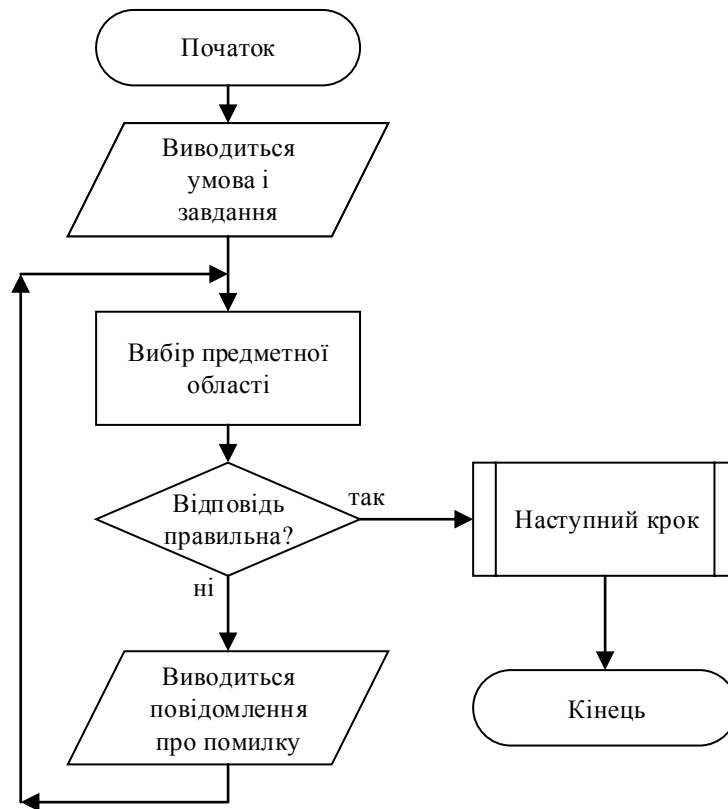


Рисунок 3.2 – Блок-схема другого кроку алгоритму

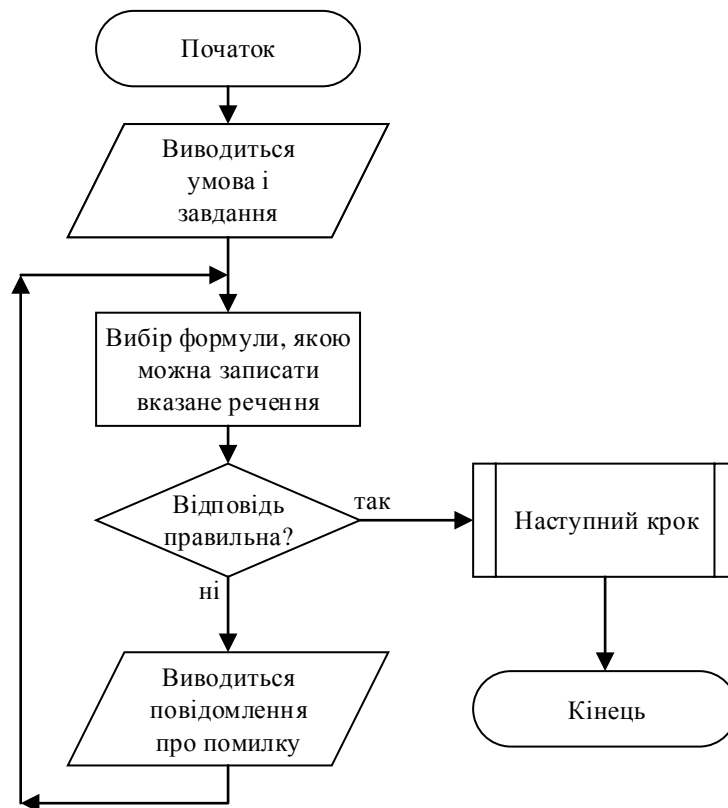


Рисунок 3.3 – Блок-схема третього кроку алгоритму

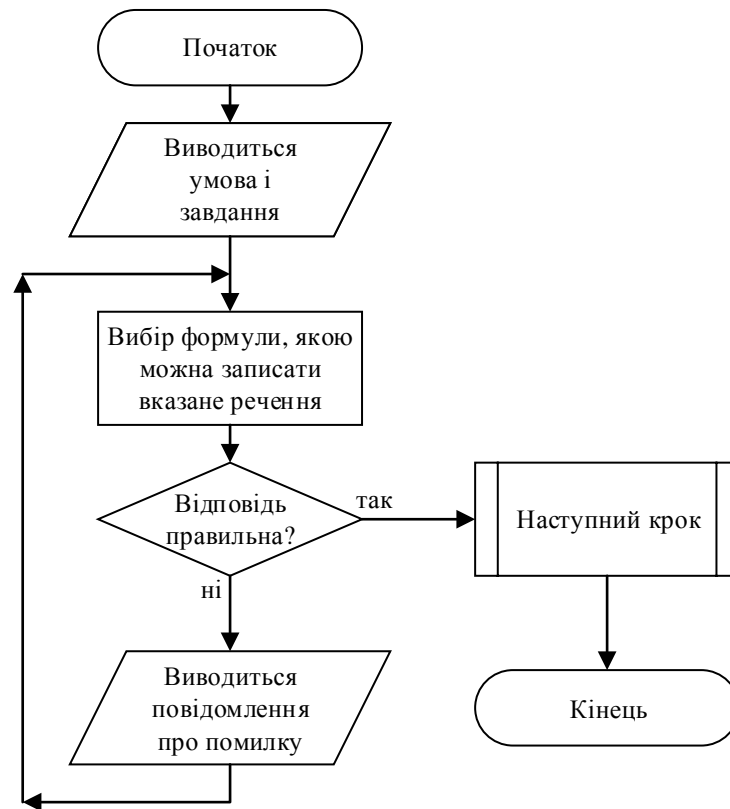


Рисунок 3.4 – Блок-схема четвертого кроку алгоритму

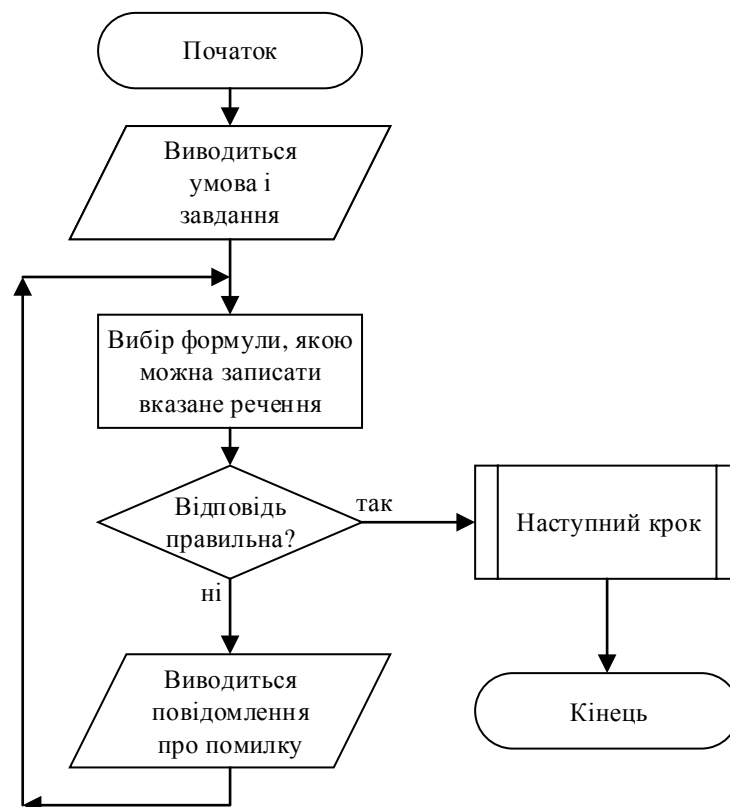


Рисунок 3.5 – Блок-схема п'ятого кроку алгоритму

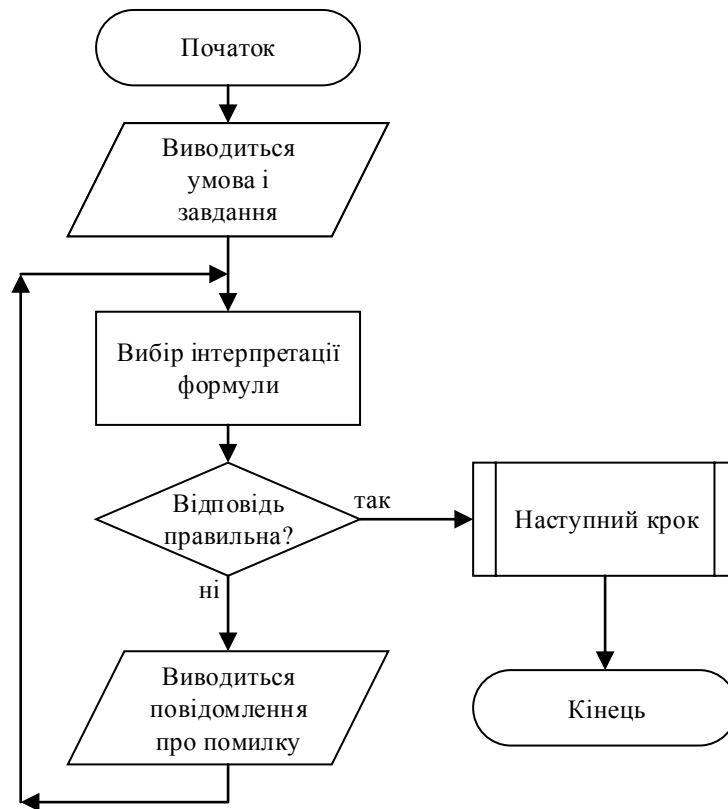


Рисунок 3.6 – Блок-схема шостого кроку алгоритму

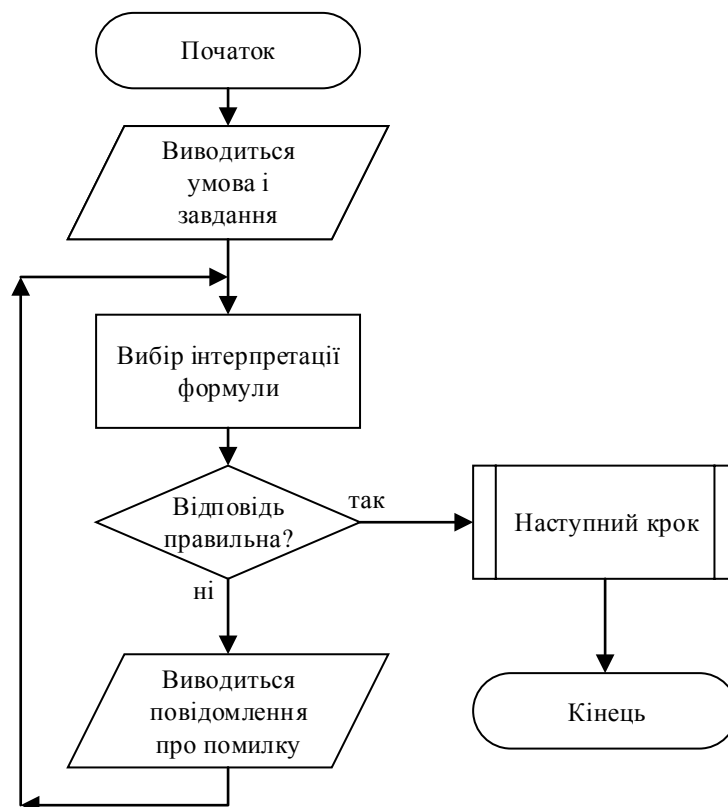


Рисунок 3.7 – Блок-схема сьомого кроку алгоритму

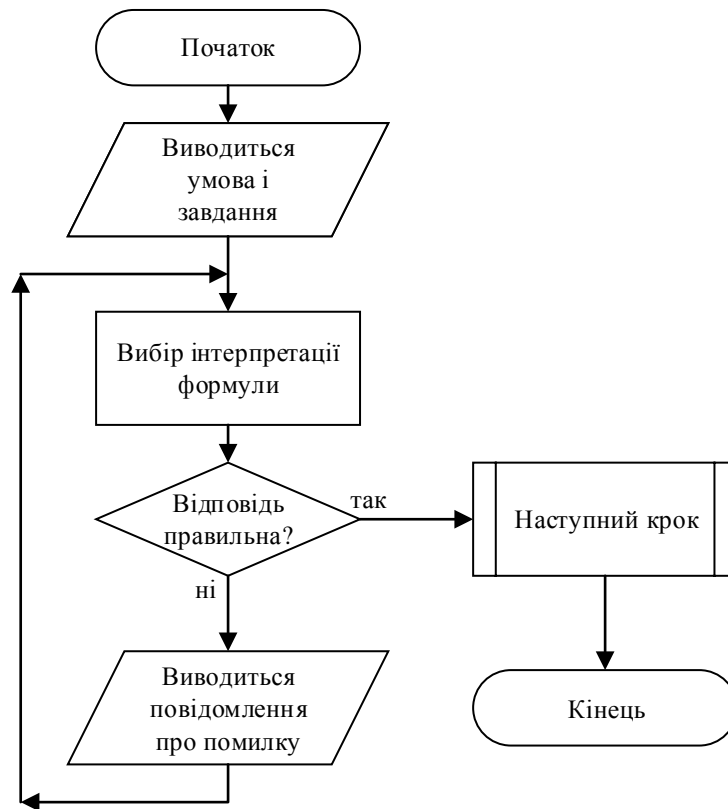


Рисунок 3.8 – Блок-схема восьмого кроку алгоритму

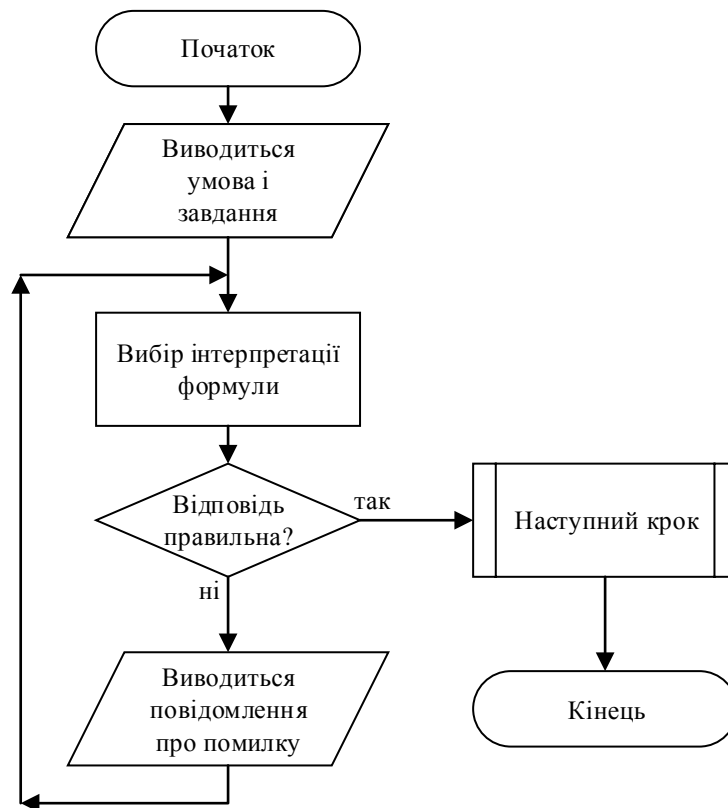


Рисунок 3.9 – Блок-схема дев'ятого кроку алгоритму

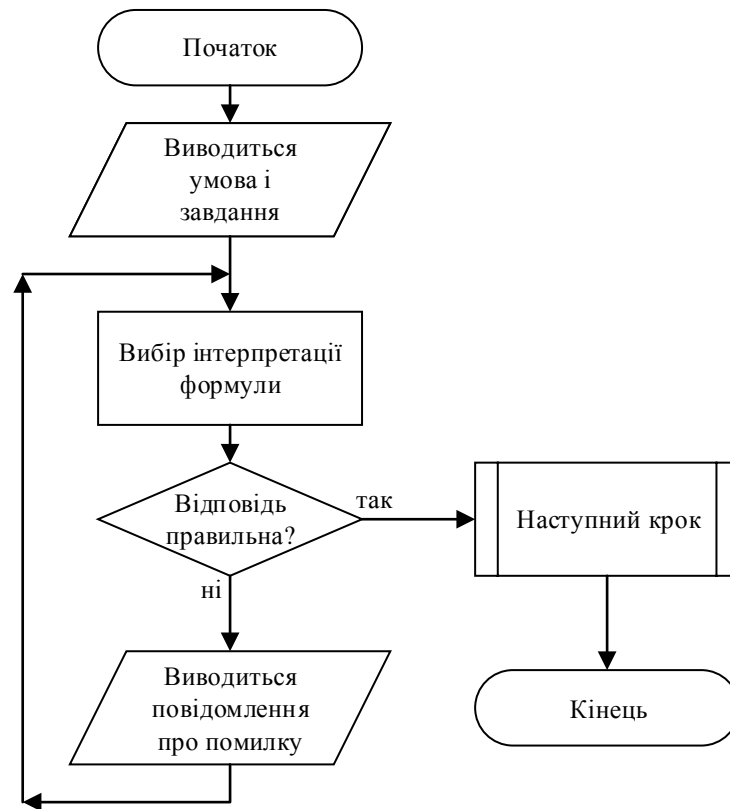


Рисунок 3.10 – Блок-схема десятого кроку алгоритму

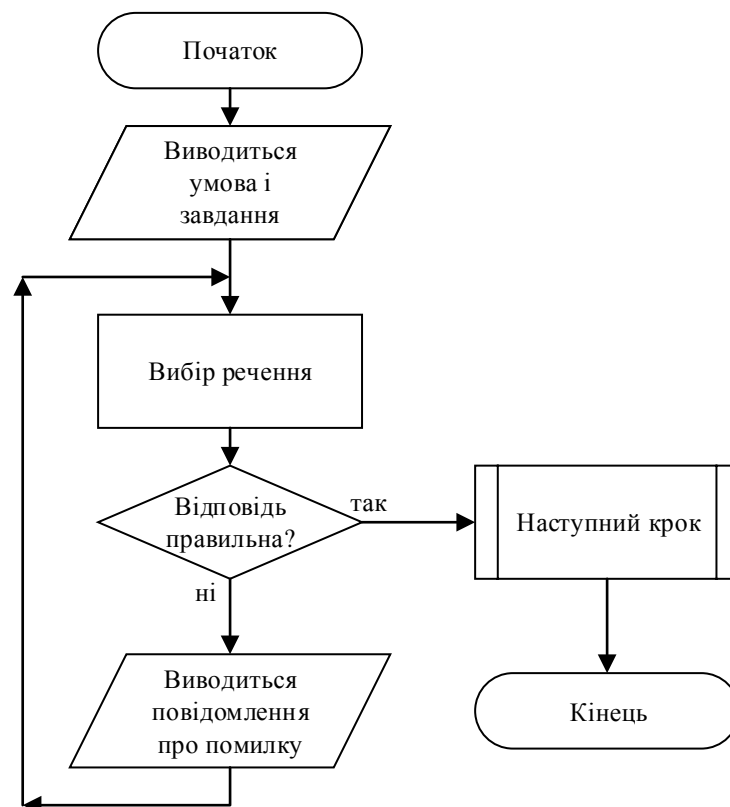


Рисунок 3.11 – Блок-схема одинадцятого кроку алгоритму

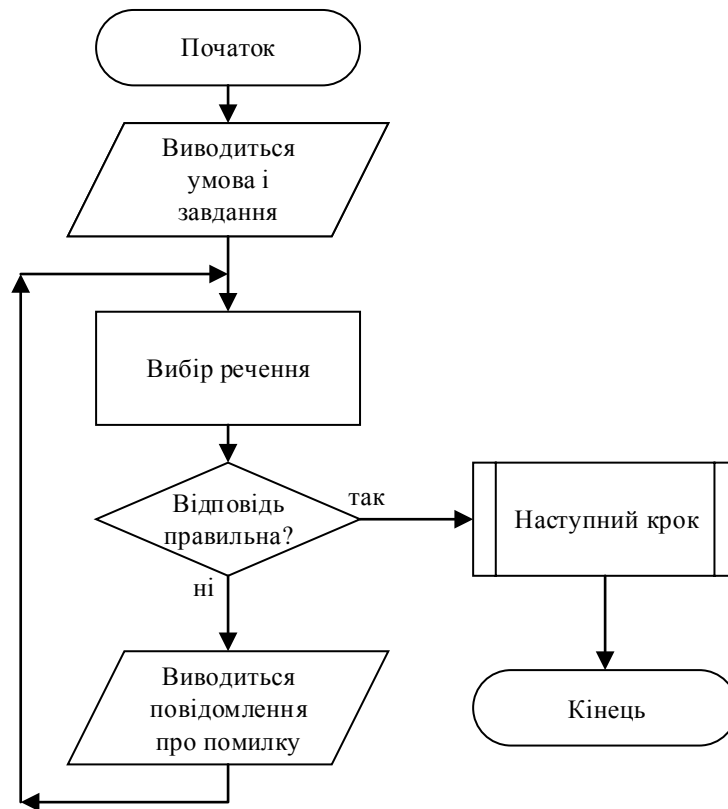


Рисунок 3.12 – Блок-схема дванадцятого кроку алгоритму

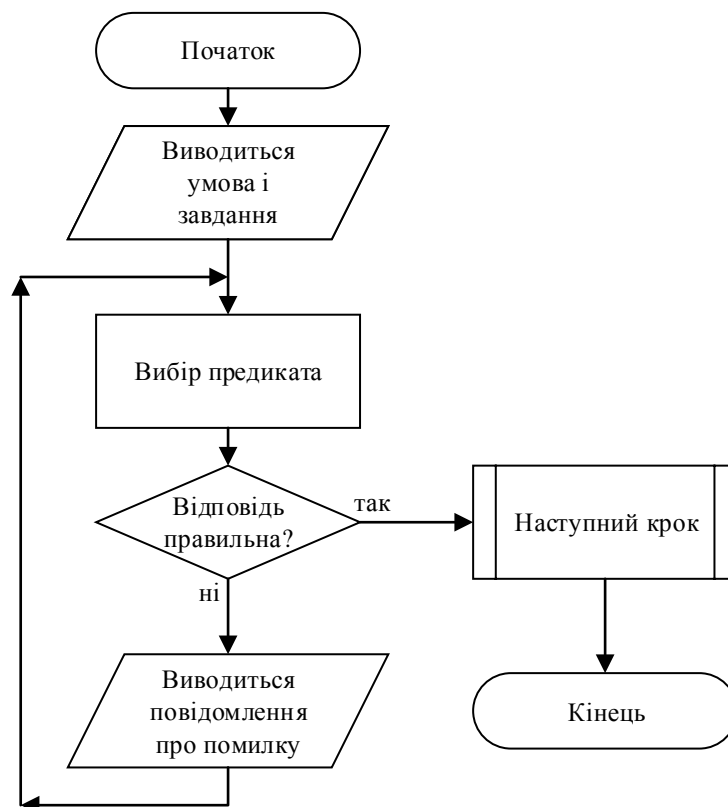


Рисунок 3.13 – Блок-схема тринадцятого кроку алгоритму

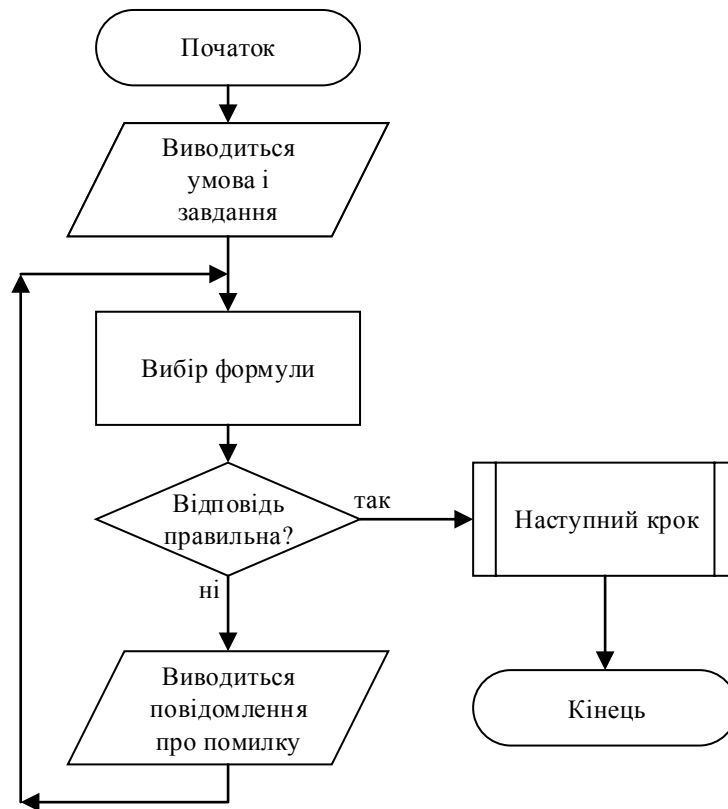


Рисунок 3.14 – Блок-схема чотирнадцятого кроку алгоритму

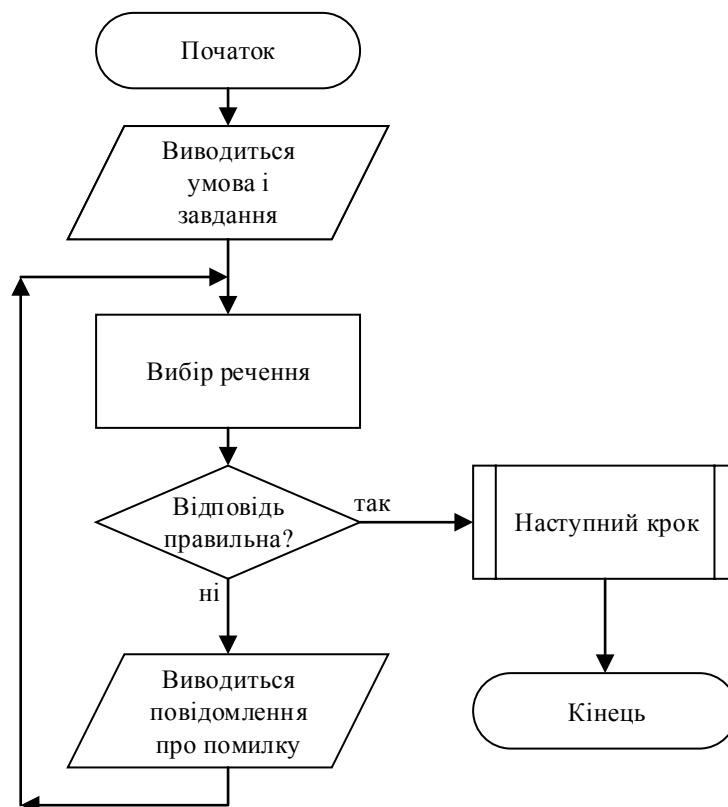


Рисунок 3.15 – Блок-схема першого кроку алгоритму





Рисунок 3.16 – Блок-схема шістнадцятого кроку алгоритму

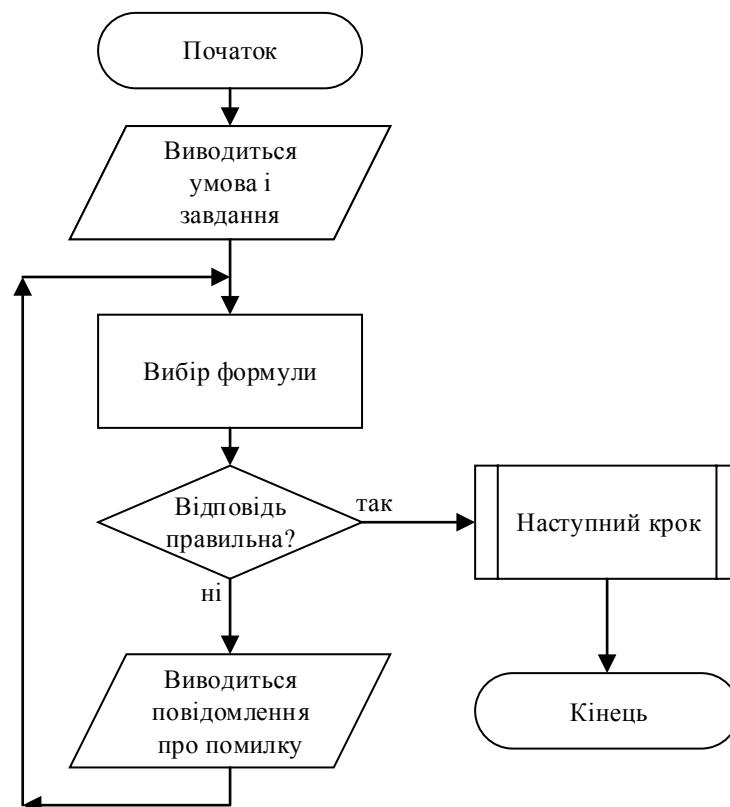


Рисунок 3.17 – Блок-схема сімнадцятого кроку алгоритму

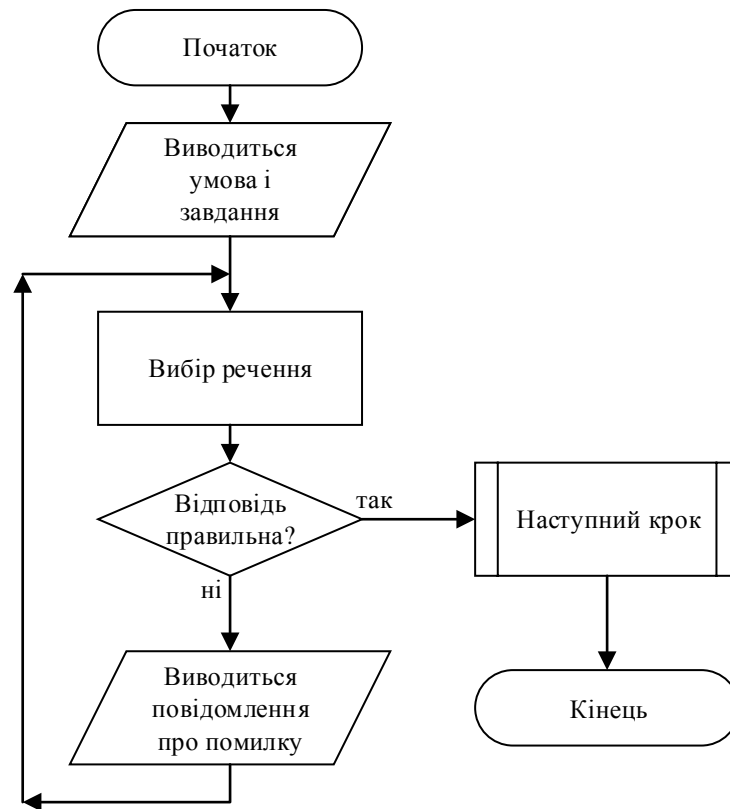


Рисунок 3.18 – Блок-схема вісімнадцятого кроку алгоритму

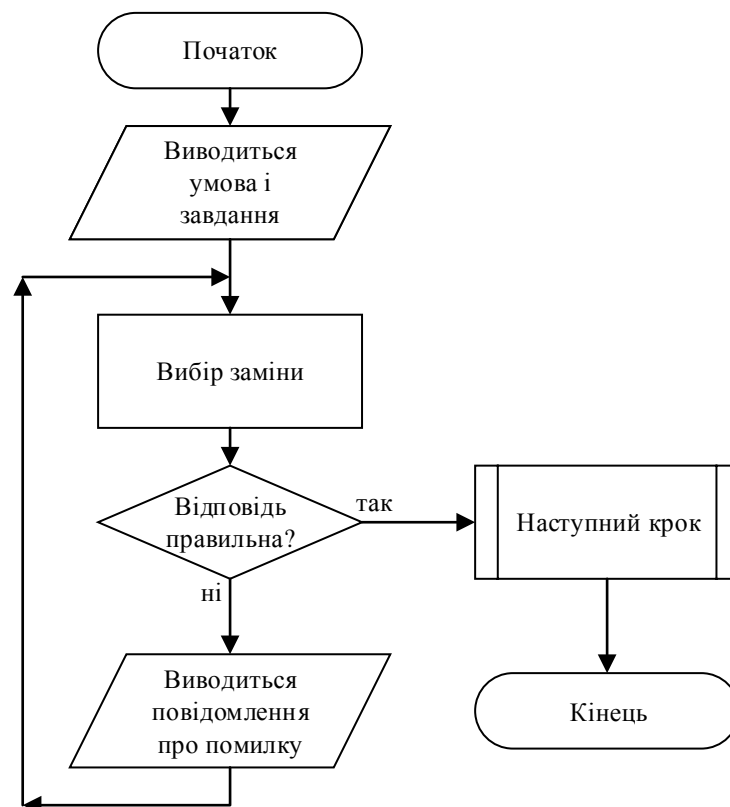


Рисунок 3.19 – Блок-схема дев'ятнадцятого кроку алгоритму

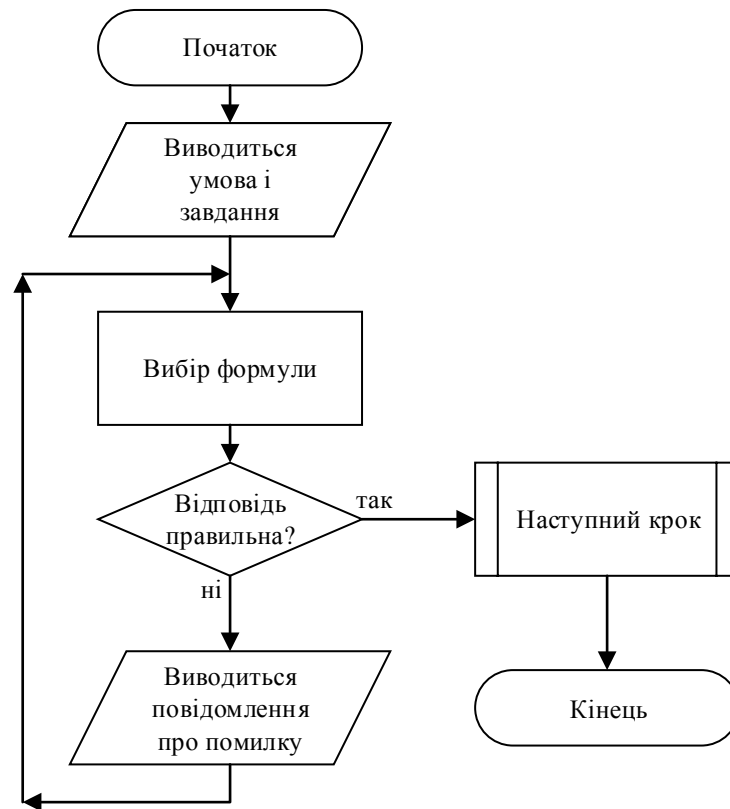


Рисунок 3.20 – Блок-схема двадцятого кроку алгоритму

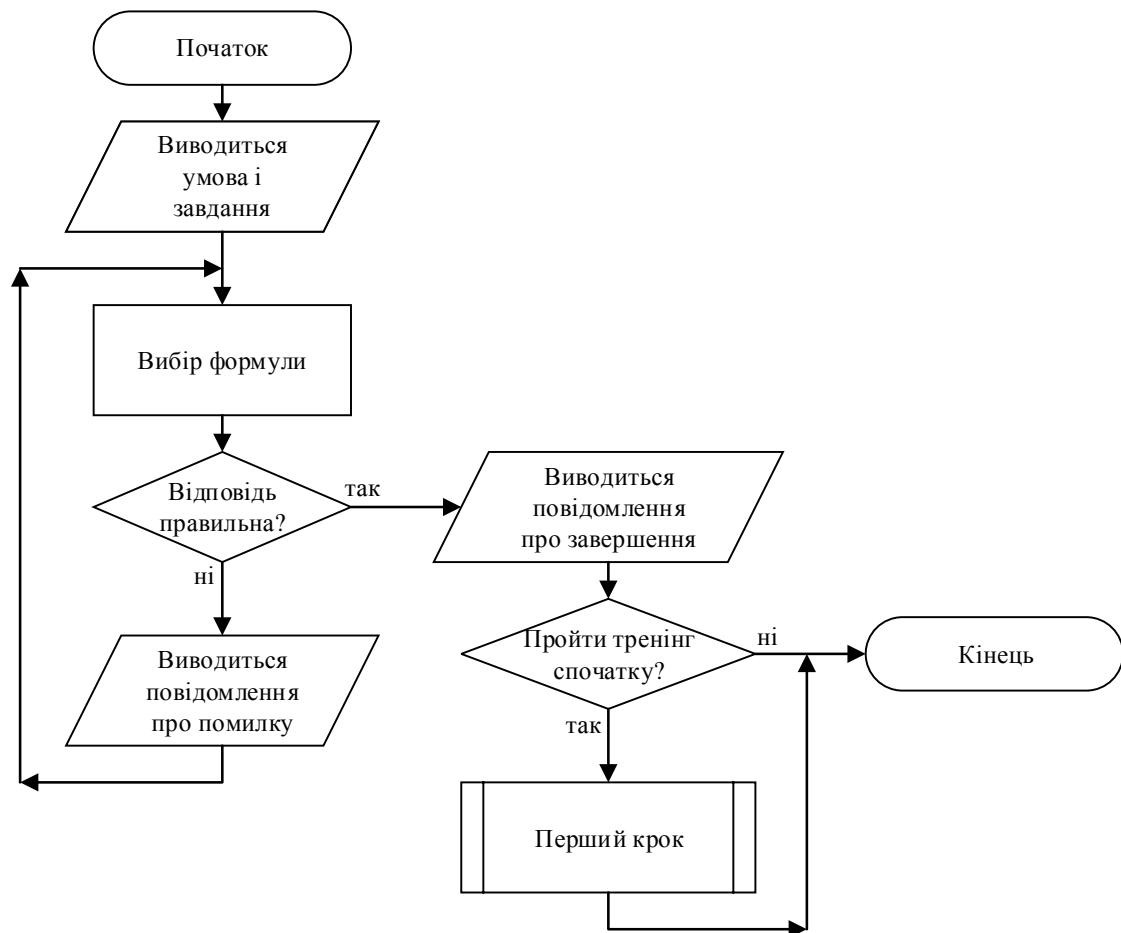


Рисунок 3.21 – Блок-схема двадцять першого кроку алгоритму

## 4. ПРАКТИЧНА ЧАСТИНА

### 4.1. Обґрунтування вибору програмних засобів

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосунків (React, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним мовам, — функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання (closures) — що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою C має такі корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів;
- функції як об'єкти першого класу;
- обробка винятків;
- автоматичне приведення типів;
- автоматичне збирання сміття;
- анонімні функції.

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JavaScript містить набір вбудованих операцій, які, грубо кажучи, не обов'язково є функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто, зрештою, успадкований від C), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, приміром, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може стояти в тексті програми після неї.

Семантика мови схожа з семантикою мови Self.

При розробці великих і нетривіальних веб-застосунків з використанням JavaScript, критично важливим є доступ до інструментів відлагодження. Оскільки браузері, від різних виробників, дещо відрізняються у поведінці JavaScript і реалізації Об'єктної моделі документа, необхідно мати відлагоджувач для кожного браузера, якщо веб-застосування орієнтовано на нього.

На даний час Firefox, Opera, Google Chrome, Edge та Safari мають зневаджувачі для себе [11].

Сучасний фреймворк AngularJS підходить для вирішення багатьох завдань у сфері веб-розробки і має багато корисних особливостей і можливостей.

1. Зв'язок даних між виведенням користувачеві і внутрішньою частиною програми.

Не потрібно ніяких ручних оновлень, щоб ознайомити користувача з даними. Зв'язок Data-binding грає велику роль і дуже корисна для розробників і користувачів. Angular самостійно автоматично оновлює всі зміни і виводить їх на екран.

2. Скорочення часу на процес написання коду.

Кожен веб-розробник як ніхто цінує свій час. Використання даного фреймворка дозволяє швидше написати потрібний код. Наприклад, Angular пов'язує дані за замовчуванням.

3. Використання html5 для додавання і створення нового функціоналу.

Використовуються саме нові стандарти html5. У підсумку, створюється поділ між файлами, які відображають контент, і безпосередньо логікою програми.

#### 4. Початок розробки - простіше простого.

Хочете почати роботу з Angular? Тоді вам варто тільки включити файл `angular.js`, який має весь необхідний базовий функціонал. Додаткові інструменти поділяються на окремі модулі, які в разі потреби можна підключати.

#### 5. Управління DOM через вбудовані директиви і атрибути.

Дані атрибути є частиною `html5`.

#### 6. Можливість створення односторінкових додатків.

Популярність таких проектів останнім часом зростає. За ідеї такі ресурси знижують навантаження на сервер і надають набагато більше можливостей для користувача. Також такі технології використовують для створення гібридних додатків для мобільних пристроїв.

#### 7. Поділ програмної і візуальної частини.

Інформація, яку Angular виводить на екран з'являється через змінні `{{param}}` і в той час, як йде розробка логіки, візуальна частина може доопрацьовуватися дизайнерами.

#### 8. Можливість вести паралельну розробку на одному ресурсі.

Дизайн фреймворка розрахований на модулярну розробку. Всі частини проекту можуть бути розділені і використані в різних комбінаціях. Цей принцип відмінно підходить в разі, коли кілька програмістів працюють над одним проектом і використовують деякі модулі повторно. Фреймворк гарне рішення для великих проектів, які збираються воєдино на завершальній етапі розробки.

#### 9. Величезне співтовариство розробників.

Це обумовлено популярністю фреймворка. Існує велика кількість форумів, де можна отримати відповіді на питання, що цікавлять по фреймворку [12, 13].

## 4.2. Опис процесу програмної реалізації

Так як тренажер реалізований через Angular CLI, то ініціалізація програми і відображення відбувається за допомогою наступного коду.

```
import {Component, OnInit} from '@angular/core';

@Component({
  selector: 'app-start-page',
  templateUrl: './start-page.component.html',
  styleUrls: ['./start-page.component.scss']
})

export class StartPageComponent implements OnInit {
  constructor() {

  }

  ngOnInit() {}
}
```

Наступним кроком є підключення плагіну для мультимовності.

```
import {Component, OnInit} from '@angular/core';
import {TranslateService} from '@ngx-translate/core';

@Component({
  selector: 'app-start-page',
  templateUrl: './start-page.component.html',
  styleUrls: ['./start-page.component.scss']
})

export class StartPageComponent implements OnInit {
  constructor(public translate: TranslateService) {
    translate.setDefaultLang('ua');
  }

  ngOnInit() {

  }

  useLanguage(language: string) {
    this.translate.use(language);
  }
}
```

`import {TranslateService} from '@ngx-translate/core'` – використовується для імпорту плагіна у компонент.

`translate.setDefaultLang('ua')` – задає українську мову по замовчуванню.



`useLanguage(language: string) { this.translate.use(language); }` – функція для зміни мови.

Для перекладу використовуються спеціальні ключі, які задаються разом з перекладом і зберігаються у json файлі.

Для української:

```
"FIRST_QUESTION": "Нехай предикат  $P(x)$  відповідає реченню « $x \geq 1$ ». Вказати предметну область, при якій висловлювання  $\forall x P(x)$  істинне.",
"FIRST_QUESTION_ANSWER_FIRST": "Предметна область – множина дійсних чисел",
"FIRST_QUESTION_ANSWER_SECOND": "Предметна область – множина натуральних чисел",
"FIRST_QUESTION_ANSWER_THIRD": "Предметна область – множина цілих чисел",
"FIRST_QUESTION_ANSWER_FOURTH": "Всі відповіді вірні",
```

Для англійської:

```
"FIRST_QUESTION": "Let predicate  $P(x)$  correspond to the sentence « $x \geq 1$ ». Specify the subject area in which the expression  $\exists x P(x)$  is true.",
"FIRST_QUESTION_ANSWER_FIRST": "The subject area is the set of real numbers",
"FIRST_QUESTION_ANSWER_SECOND": "The subject area is the set of natural numbers",
"FIRST_QUESTION_ANSWER_THIRD": "The subject area is the set of integers",
"FIRST_QUESTION_ANSWER_FOURTH": "All answers are correct",
```

Великим плюсом цього способу є те, що розробник може віддати ці json файли, любій другій людині, і вона може змінити текст в них як в звичайному блокноті, а потім повернути розробнику.

Сам текст виводиться так:

```
{{'CONTENT.FIRST_QUESTION_ANSWER_FIRST | translate}}}
```

Самі питання також задаються через json:

```
"quest": [
  {
    "id": 1,
    "question": "CONTENT.FIRST_QUESTION",
    "answers": [
      {
        "text": "CONTENT.FIRST_QUESTION_ANSWER_FIRST",
        "isRight": false
      },
      {
        "text": "CONTENT.FIRST_QUESTION_ANSWER_SECOND",
        "isRight": true
      },
      {
        "text": "CONTENT.FIRST_QUESTION_ANSWER_THIRD",
        "isRight": false
      },
    ],
  },
]
```

```

    {
      "text": "CONTENT.FIRST_QUESTION_ANSWER_FOURTH",
      "isRight": false
    }
  ]

```

Quest-це масив об'єктів, в яких знаходяться запитання.

Id- це индекс запитання.

Question – саме запитання.

Answers – також масив об'єктів, з відповідями.

Text – варіант відповіді.

isRight – якщо true, то відповідь правильна.

Потім було розроблено сам дизайн стартової сторінки, так компонент start-page має: інформацію про тренажер, кнопки для переходу до теорії, практики на англійській мові та українській, також є select з вибором мови.

Код HTML сторінки:

```

<div class="lang-select">
  <mat-form-field>
    <mat-label>{{'CONTENT.CHOOSE_LANGUAGE' | translate}}</mat-label>
    <mat-select>
      <mat-option (click)="useLanguage('ua')" value="ua">Укр</mat-
option>
      <mat-option (click)="useLanguage('en')" value="eng">Eng</mat-
option>
    </mat-select>
  </mat-form-field>
</div>
<div class="start-page">
  <div>
    <p>{{'CONTENT.SIMULATOR' | translate}}</p>
    <p class="bold">{{'CONTENT.HOME_TITLE' | translate}}</p>
    <p>{{'CONTENT.DISCIPLINE' | translate}}</p>
  </div>
  <div class="start-page__creator">
    <div class="flex align-items-center justify-content-center">
      <p class="bold">{{'CONTENT.CREATOR' | translate}}</p>
      <p>&nbsp;{{'CONTENT.ME' | translate}}</p>
    </div>

    <div class="flex align-items-center justify-content-center">
      <p class="bold">{{'CONTENT.TEACHER' | translate}}</p>
      <p>&nbsp;{{'CONTENT.TEACHER_NAME' | translate}}</p>
    </div>
    <button class="start-page__theory-button" mat-raised-button
color="primary"
      routerLink="/theory-page">{{'CONTENT.THEORY_BUTTON' |
translate}}</button>
  </div>
  <div class="start-page__button-container">

```

```

        <button mat-raised-button color="primary"
(click)="useLanguage('en') "
            routerLink="/main-page">{{ 'CONTENT.ENG_BUTTON' |
translate}}</button>
        <button mat-raised-button color="primary"
(click)="useLanguage('ua') "
            routerLink="/main-page">{{ 'CONTENT.UA_BUTTON' |
translate}}</button>
    </div>
</div>

```

### Ta SCSS:

```

.start-page {
    font-size: 24px;
    text-align: center;
    height: calc(100% - 70px);
    display: flex;
    flex-direction: column;
    justify-content: center;
    transform: translateY(-5%);

    &__button-container {
        margin-top: 50px;

        button {
            margin: 10px;
        }
    }

    &__creator {
        p {
            margin: 0;
        }
    }

    &__theory-button {
        margin-top: 20px;
    }
}

.lang-select {
    text-align: right;
    position: relative;
    z-index: 1;
}

```

Також була застосована методологія BEM(Block-Element-Modifier), та бібліотека angular material для дизайну.

Сторінка з запитаннями знаходиться у компоненті main-page. На якій знаходяться: назва тренажера, запитання, варіанти відповіді, кнопка далі(ящо це не останнє питання), назад(якщо це на перше запитання), та спочатку, яка повертає на стартову сторінку, а також панель на якій

відображенню на якому кроці знаходиться користувач. Нижче представлення HTML сторінки:

```
<mat-toolbar color="primary">
  <mat-toolbar-row>
    {{'CONTENT.HOME_TITLE' | translate}}
  </mat-toolbar-row>
</mat-toolbar>
<div class="main-page">
  <mat-horizontal-stepper linear #stepper>
    <mat-step [optional]="isOptional" [stepControl]="firstFormGroup"
*ngFor="let step of question; let first = first; let last = last;
let questIndex = index">
      <ng-template matStepLabel>{{'CONTENT.STEP' | translate}}</ng-
template>
      <form [formGroup]="firstFormGroup">
        <label id="example-radio-group-label">{{step.question |
translate}}</label>
        <mat-radio-group
          FormControlName="firstCtrl" required
          aria-labelledby="example-radio-group-label"
          class="example-radio-group">
          <mat-radio-button required class="example-radio-button"
*ngFor="let answer of step.answers;"
            [value]="answer.isRight">
            {{answer.text | translate}}
          </mat-radio-button>
        </mat-radio-group>
        <div class="navigation-buttons-container">
          <button mat-raised-button color="warn"
routerLink="">{{'CONTENT.AGAIN' | translate}}</button>
          <button mat-raised-button *ngIf="!first" color="accent"
matStepperPrevious>{{'CONTENT.BACK' |
            translate}}
          </button>
          <button mat-raised-button color="primary" *ngIf="!last"
(click)="checkAnswer(stepper, questIndex)">
            {{'CONTENT.NEXT' | translate}}
          </button>
        </div>
      </form>
    </mat-step>
  </mat-horizontal-stepper>
</div>
```

### SCSS:

```
.main-page {
  max-width: 1920px;
  margin: auto;
}

.navigation-buttons-container {
  button {
    margin-right: 10px;
  }
}

.example-radio-group {
  display: flex;
  flex-direction: column;
```

```

    margin: 15px 0;
}

.example-radio-button {
    margin: 5px;
}

```

## TS:

```

import {Component, OnInit} from '@angular/core';
import {FormBuilder, FormGroup, Validators} from '@angular/forms';
import {HttpParams, HttpClient} from '@angular/common/http';
import {MatDialog, MatDialogRef, MAT_DIALOG_DATA} from
'@angular/material/dialog';
import {DefaultDialogComponent} from '../shared/default-dialog/default-
dialog.component';

@Component({
    selector: 'app-main-page',
    templateUrl: './main-page.component.html',
    styleUrls: ['./main-page.component.scss']
})
export class MainPageComponent implements OnInit {
    firstFormGroup: FormGroup;
    secondFormGroup: FormGroup;
    isOptional = false;

    question: any[] = [];

    constructor(private _formBuilder: FormBuilder,
                private http: HttpClient,
                public dialog: MatDialog) {

    }

    checkAnswer(stepper: any, currentAnswer) {
        // console.log(this.question[b].map(item));
        const rightAnswer =
this.question[currentAnswer].answers.filter(item => item.isRight === true);
        if (this.firstFormGroup.valid) {
            if (this.firstFormGroup.value.firstCtrl) {
                stepper.next();
            } else {
                const dialogRef = this.dialog.open(DefaultDialogComponent,
{
                    data: rightAnswer[0].text
                });

                dialogRef.afterClosed().subscribe(result => {
                });
            }
        } else {
            const dialogRef = this.dialog.open(DefaultDialogComponent, {
                data: 'CONTENT.NO_ANSWER'
            });

            dialogRef.afterClosed().subscribe(result => {
            });
        }
    }

    ngOnInit() {

```

```

    this.firstFormGroup = this._formBuilder.group({
      firstCtrl: ['', Validators.required]
    });
    this.secondFormGroup = this._formBuilder.group({
      secondCtrl: ['', Validators.required]
    });

    this.http.get('assets/data/data.json').subscribe((data: any) => {
      console.log(data);
      this.question = data.quest;
    });
  }
}

```

Функція `checkAnswer` перевіряє правильність відповіді користувача, якщо він обирає правильну відповідь, він переходить до наступного запитання, якщо нічого не обрав, або не правильно, то отримує повідомлення з відповідним текстом.

За повідомлення відповідає окремий компонент `default-dialog`.

HTML:

```
{{data | translate}}
```

Також була реалізована адаптивність під різні екрани: мобільні, планшетні та десктопні.

### 4.3. Опис програми

Головна сторінка містить назву тренажеру, інформацію про розробника і керівника (рис. 4.1).

Кнопки «Продовжити на англійській» та «Продовжити на українській» пропонує користувачу перейти до практики на відповідній мові.

У вкладці «Переглянути теорію» розміщено теоретичний матеріал, наведено приклади (рис. 4.2).

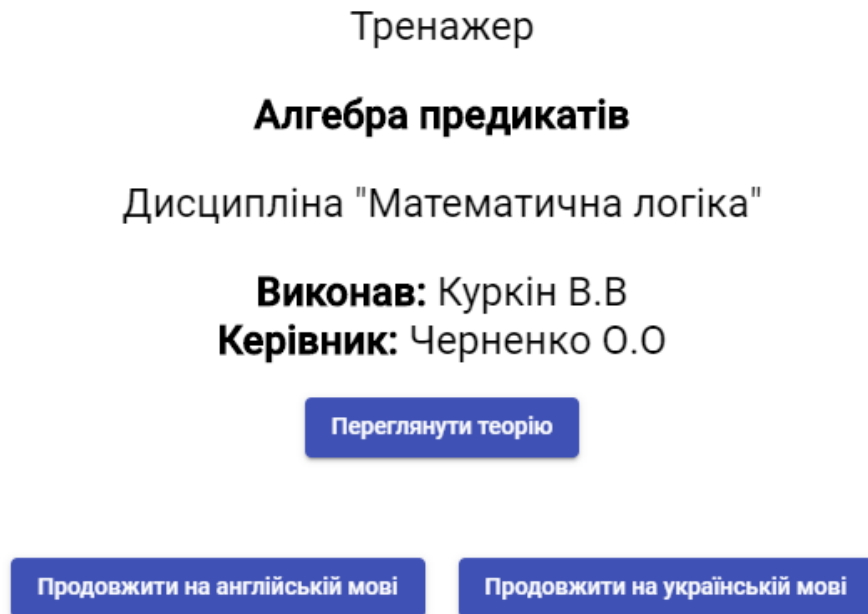


Рисунок 4.1 – Головна сторінка тренажеру

## Приклади алгебри предикатів

### Приклад 1.

Навести приклади інтерпретацій формули алгебри предикатів  $\forall x(P(x)) \rightarrow Q(x)$ .

### Розв'язання.

1. Область інтерпретації – множина живих істот,  $P(x)$  :  $x$  – риба,  $Q(x)$  :  $x$  живе у воді. Інтерпретація формули: «Всі риби живуть у воді».
2. Область інтерпретації – множина живих істот,  $P(x)$  :  $x$  – людина,  $Q(x)$  :  $x$  смертний. Інтерпретація формули: «Всі люди смертні».
3. Область інтерпретації – множина цілих чисел,  $P(x)$  :  $x$  ділиться на 6,  $Q(x)$  :  $x$  ділиться на 3. Інтерпретація формули: «Всі числа, які діляться на 6, діляться на 3».

### Приклад 2.

Нехай предикат  $P(x)$  відповідає реченню « $x \geq 1$ », а предметна область складається з усіх дійсних чисел. Тоді висловлювання  $\forall x(P(x))$  хибне:  $\forall x(P(x)) = F$ .

Якщо ж предмета область складається з усіх натуральних чисел, то висловлювання  $\forall x(P(x))$  істинне:  $\forall x(P(x)) = T$ .

### Приклад 3.

Позначимо речення « $x$  – просте число» як  $P(x)$ , « $x$  – раціональне число» – як  $Q(x)$ , « $x$  – дійсне число» як  $R(x)$  та « $x$  менше  $y$ » – як  $МЕНШЕ(x, y)$ . Розглянемо такі істинні твердження.

1. Кожне раціональне число дійсне.
2. Існує просте число.
3. Для кожного числа  $x$  існує таке число  $y$ , що  $x < y$ .

Наведені речення можна записати такими формулами.

1.  $\forall x(P(x)) \rightarrow Q(x)$
2.  $\exists x(P(x))$
3.  $\forall x \exists y МЕНШЕ(x, y)$ .

Рисунок 4.2 – Приклад вкладки «Теорія»

Запитання наводяться з вибором одного з варіантів відповіді, а також надається можливість повернутися на початок програми (рис. 4.3).

**Алгебра предикатів**

1 Крок — 2 Крок — 3 Крок — 4 Крок — 5 Крок — 6 Крок —

Нехай предикат  $P(x)$  відповідає реченню « $x \geq 1$ ». Вказати предметну область

☐ Предметна область – множина дійсних чисел  
☐ Предметна область – множина натуральних чисел  
☐ Предметна область – множина цілих чисел  
☐ Всі відповіді вірні

Спочатку Далі

Рисунок 4.3 – Запитання з вибором відповіді

Якщо відповіді неправильно на будь-яке завдання, то виведеться повідомлення про помилку з вірною відповіддю (рис. 4.4).

1 Крок — 2 Крок — 3 Крок — 4 Крок — 5 Крок — 6 Крок — 7 Крок — 8 Крок — 9 Крок — 10 Крок — 11 Крок — 12 Крок — 13 Крок — 14 Крок — 15 Крок —

Нехай предикат  $P(x)$  відповідає реченню « $x \geq 1$ ». Вказати предметну область, при якій висловлювання  $\forall x P(x)$  істинне.

☐ Предметна область – множина дійсних чисел  
☐ Предметна область – множина натуральних чисел  
☒ Предметна область – множина цілих чисел  
☐ Всі відповіді вірні

Спочатку Далі

Предметна область – множина натуральних чисел

Рисунок 4.4 – Повідомлення про помилку



Якщо не обрати відповідь, то відобразиться повідомлення про те, що слід обрати відповідь (рис. 4.5).

1 Крок — 2 Крок — 3 Крок — 4 Крок — 5 Крок — 6 Крок — 7 Крок — 8 Крок — 9 Крок — 10 Крок — 11 Крок — 12 Крок — 13 Крок — 14

Нехай предикат  $P(x)$  відповідає реченню « $x \geq 1$ ». Вказати предметну область, при якій висловлювання  $\forall x P(x)$  істинне.

☐ Предметна область – множина дійсних чисел  
☐ Предметна область – множина натуральних чисел  
☐ Предметна область – множина цілих чисел  
☐ Всі відповіді вірні

Спочатку Далі

Треба обрати відповідь

Рисунок 4.5 – Повідомлення про помилку

При завершенні відображається відповідне повідомлення, а також надається можливість повернутися на початок (рис. 4.6).

## Алгебра предикатів

1 Крок — 2 Крок — 3 Крок — 4 Крок — 5 Крок — 6 Крок — 7 Крок — 8 Крок

Вітаємо, ви закінчили проходження тренажеру.

Спочатку Назад

Рисунок 4.6 – Повідомлення про завершення

#### 4.4. Інструкція по використанню тренажеру

Після запуску тренажера на головній сторінці виводиться назва тренажеру, інформація про розробника і керівника (рис. 4.1). Натиснувши кнопку «Переглянути теорію» відобразиться теоретичний матеріал з даної теми (рис. 4.2).

Щоб змінити мову вмісту програми необхідно вибрати її в селекторі (рис. 4.7). По замовчуванню вибрано українську.

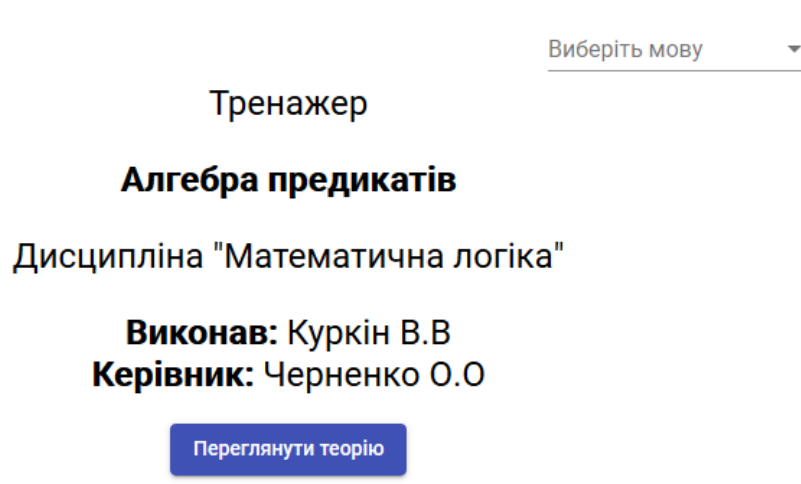


Рисунок 4.7 – Вибір мови

Якщо було обрано англійську, то головна сторінка буде виводитися на цій мові (рис. 4.8). Це стосується і теоретичного матеріалу (рис. 4.9).

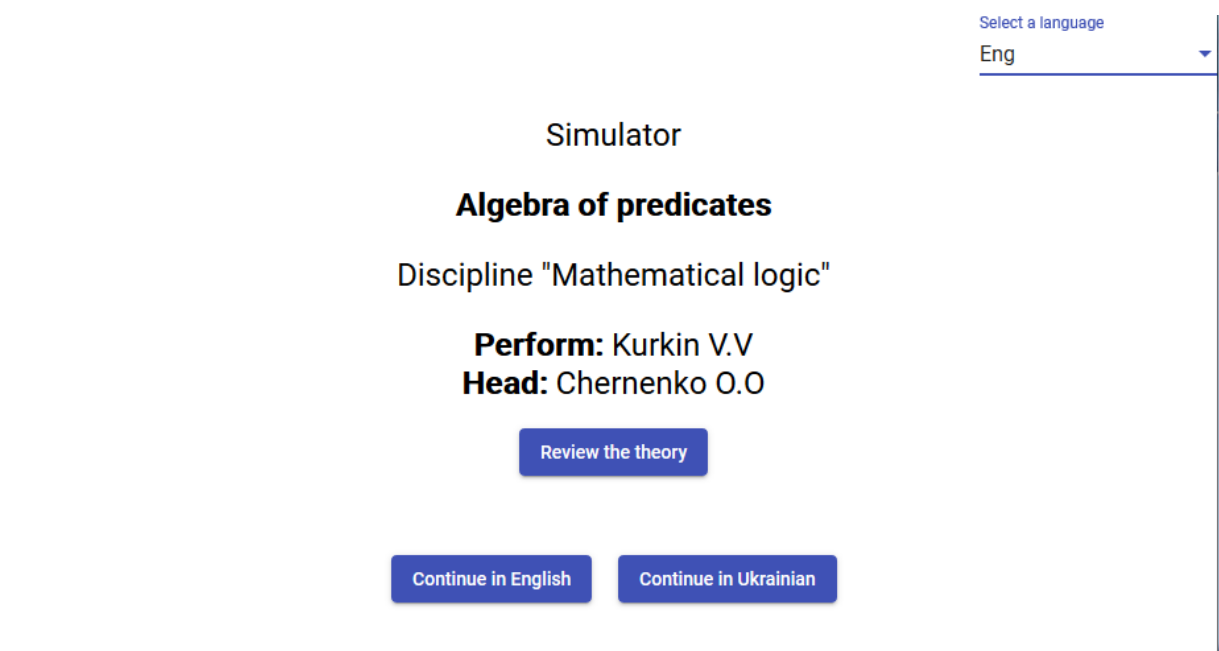


Рисунок 4.8 – Головна сторінка тренажеру англійською мовою

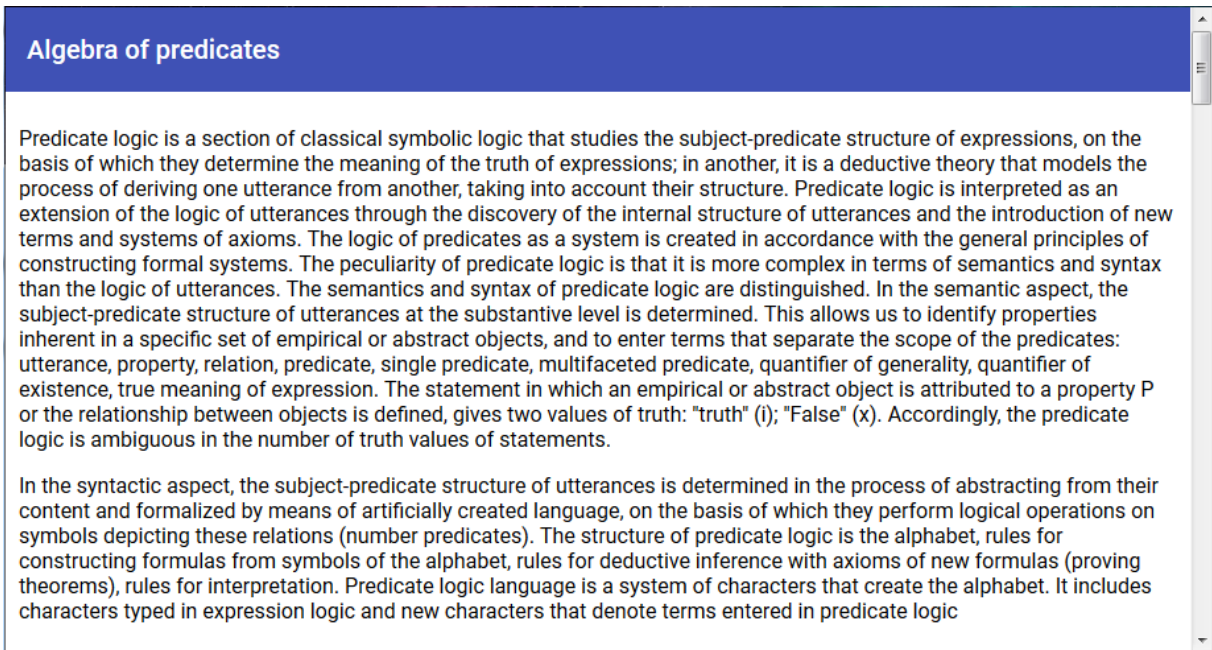


Рисунок 4.9 – Приклад вкладки «Теорія» англійською мовою

При переході до практики також пропонується якою мовою продовжити – кнопки «Продовжити на англійській мові» і «Продовжити на українській мові».

Після цього наводиться перше запитання з вибором одного з варіантів відповіді (рис. 4.3). Якщо не обрати відповідь, то відобразиться відповідне повідомлення (рис. 4.5).

Натиснувши кнопку «Далі» перевіряється вибрана відповідь.

Якщо відповісти неправильно, то виведеться повідомлення про помилку (рис. 4.4). Щоб закрити помилку потрібно ще раз натиснути в будь-якому місці тренажера.

В іншому разі – виведеться наступне запитання та варіанти відповіді (рис. 4.10).

**Алгебра предикатів**

Нехай предикат  $P(x)$  відповідає реченню « $x \geq 1$ ». Вказати предметну область, при якій висловлювання  $\exists x P(x)$  істинне.

- ☐ Предметна область – множина дійсних чисел
- ☐ Предметна область – множина натуральних чисел
- ☐ Предметна область – множина цілих чисел
- ☐ Всі відповіді вірні

Спочатку Назад Далі

Рисунок 4.10 – Друге запитання

Відповівши на всі запитання виводиться повідомлення про завершення, а також надається можливість повернутися на початок. Для цього потрібно натиснути кнопку «Спочатку» (рис. 4.6).

## ВИСНОВКИ

Роботу виконано з наведеними актуальністю обраної теми, оглядом переваг і недоліків дистанційного навчання, оглядом тренажерів і їх особливостей, алгоритмом роботи тренажера, блок-схемою алгоритму.

Основні результати роботи:

- вказано актуальність теми;
- розглянуто теоретичні відомості про алгебру предикатів, логіку першого порядку;
- розглянуто специфіку застосування алгебри предикатів;
- розроблено алгоритм тренажера з теми «Алгебра предикатів»;
- створено програмну реалізацію алгоритму;
- описано процес створення програми;
- зроблено опис розробленої програми;
- написано інструкцію користувача.

В алгоритмі розроблено можливість студента звернутися до теоретичного матеріалу з теми, що допоможе йому у проходженні завдання та покроково описано всі дії при розв'язанні прикладу.

Заочна форма освіти забезпечує:

- можливість отримання освіти незалежно від місця проживання;
- одночасного навчання за декількома спеціальностями;
- можливість отримання вищої освіти різними категоріями населення (непрацездатні, інваліди, жінки, які виховують дітей тощо);
- одночасне охоплення великої кількості студентів;
- навчання за власним графіком у міжсесійний період;
- можливість практично негайно застосовувати свої знання на практиці, не закінчивши навчання.

Тренажер з теми «М-метод для задач лінійного програмування» дистанційного навчального курсу «Методи оптимізації та дослідження

операцій» імітує процес розв'язування задачі лінійного програмування М-методом на основі заданого прикладу. В свою чергу тренажер з теми «Системи числення» дистанційного навчального курсу «Інформаційні мережі» навчає студентів системам числення і основним операціям в них

Так як тренажер реалізований через Angular CLI, то спочатку відбувається ініціалізація програми і відображення. Наступним кроком є підключення плагіну для мультязичності.

Для перекладу використовуються спеціальні ключі, які задаються разом з перекладом і зберігаються у json файлі. Великим плюсом цього способу є те, що розробник може віддати ці json файли, любій другій людині, і вона може змінити текст в них як в звичайному блокноті, а потім повернути розробнику.

Самі питання також задаються через json.

Потім було розроблено сам дизайн стартової сторінки - компонент start-page. Сторінка з запитаннями знаходиться у компоненті main-page.

Функція checkAnswer перевіряє правильність відповіді користувача, Розроблений тренажер має зрозумілий для користувача інтерфейс.

Головна сторінка тренажера містить інформацію щодо назви тренінгу, його розробника та керівника. На кожному кроці виводиться умова прикладу, завдання та наводяться варіанти відповіді. Лише один з варіантів відповідей є вірним.

У вкладці «Переглянути теорію» розміщено теоретичний матеріал, наведено приклади.

Запитання наводяться з вибором одного з варіантів відповіді, а також надається можливість повернутися на початок програми. Якщо відповісти неправильно на будь-яке завдання, то виведеться повідомлення про помилку з вірною відповіддю.

Якщо не обрати відповідь, то відобразиться повідомлення про те, що слід обрати відповідь.

При завершенні відображається відповідне повідомлення, а також надається можливість повернутися на початок.

Результати роботи були опубліковані в матеріалах науково-практичного семінару «Комп'ютерні науки і прикладна математика (КНіПМ-2019)» та у Збірнику наукових статей магістрів Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі».

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання дипломної роботи для студентів ступеня магістра спеціальності 122 «Комп'ютерні науки» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2018. – 35 с.
2. Куркін В.В. Елементи програмного забезпечення для тренажера з теми «Алгебра предикатів» дистанційного навчального курсу «Математична логіка» / В.В. Куркін // Комп'ютерні науки і прикладна математика (КНіПМ-2019): матеріали науково-практичного семінару. Випуск 3 / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2019. – 83 с. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/7041>
3. Системи дистанційного навчання: огляд, аналіз, вибір [Електронний ресурс] / Б. Демида, С. Сагайдак, І. Копил // Вісник Національного університету "Львівська політехніка". – 2011. – № 694 : Комп'ютерні науки та інформаційні технології. – С. 98-107. – Режим доступу: <http://ena.lp.edu.ua/bitstream/ntb/10662/1/14.pdf>
4. Дистанційний курс «Методи оптимізації та дослідження операцій (Частина 1) (2019) // Головний центр дистанційного навчання вищого навчального закладу УКООПСПЛКИ «Полтавський університет економіки і торгівлі»». – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=1211>
5. Педоренко С.В. Розробка тренажера з теми "М-метод" дистанційного навчального курсу "Методи оптимізації та дослідження операцій" / С.В. Педоренко, О.О. Ємець // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 16–18 берез. 2017 р.). – Полтава: ПУЕТ, 2017. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/5423>
6. Дистанційний курс «Інформаційні мережі» // Головний центр дистанційного навчання вищого навчального закладу УКООПСПЛКИ



- «Полтавський університет економіки і торгівлі»». – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=1004>
7. Чверткін А.В. Разработка тренажера по теме "Системы счисления" дистанционного обучающего курса "Архитектура вычислительных систем" / А.В. Чверткін // Інформатика та системні науки (ІСН-2014) : матеріали V Всеукр.-наук.-практ. конф., (м. Полтава, 13–15 березня 2014 р.). – Полтава: ПУЕТ, 2014. – С. 314-315. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/2847>
  8. Нікольський Ю.В. Дискретна математика / Ю.В. Нікольський, В.В. Пасічник, Ю.М. Щербина. – К.: Видавнича група ВНУ, 2007. – 368 с.
  9. Таран Т.А. Основи дискретної математики / Т.А. Таран. – К.: Просвіта, 2003. – 288 с.
  10. Жук П. Ф. Математична логіка та теорія алгоритмів : практикум / уклад.: / П. Ф. Жук – К. : НАУ, 2014. – 21 с.
  11. JavaScript [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – Режим доступу: <https://uk.wikipedia.org/wiki/JavaScript>
  12. Почему стоит выбрать фреймворк AngularJS? [Електронний ресурс] // Компания HyperHost. – Режим доступу: <https://hyperhost.ua/info/pochemu-stoit-vyibrat-freymvork-angularjs/>
  13. Руководство по Angular 8 [Електронний ресурс] // METANIT.COM Сайт о программировании. – Режим доступу: <https://metanit.com/web/angular2/>
  14. Пабло Дилеман Изучаем Angular 2/ Пабло Дилеман. – ДМК-Пресс, 2017. – 356 с.
  15. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.